2022

Perceptions of Coding Instruction in K-12 Archdiocese of Los Angeles Catholic Schools

Krikor Koko Kiladjian
Loyola Marymount University

LMU/LLS Theses and Dissertations

2022

Perceptions of Coding Instruction in K-12 Archdiocese of Los Angeles Catholic Schools

Krikor Koko Kiladjian

LOYOLA MARYMOUNT UNIVERSITY

Perceptions of Coding Instruction in K–12 Archdiocese of Los Angeles Catholic Schools

by

Krikor Koko Kiladjian

A dissertation presented to the Faculty of the School of Education,

Loyola Marymount University,

in partial satisfaction of the requirements for the degree

Doctor of Education

2022

Perceptions of Coding Instruction in K–12 Archdiocese of Los Angeles Catholic Schools

by

Krikor Koko Kiladjian

Loyola Marymount University School of Education

Los Angeles, CA 90045

This dissertation written by Krikor Kiladjian, under the direction of the Dissertation Committee, is approved and accepted by all committee members, in partial fulfillment of requirements for the degree of Doctor of Education.

24 May 2022

Date

Dissertation Committee

[redacted]

Elizabe[redacted]d.D., Dissertation Chair

Philip Molebash, Ph.D., Committee Member

Peter Rich, Ph.D., Committee Member

ACKNOWLEDGMENTS

There are many people who guided and assisted me during this journey, and I would be remiss if I did not mention each of them individually. I would like to begin by expressing my sincerest gratitude to my parents. They were proud of me for making the decision to enter a doctoral program at that point in my life and were constantly supportive during the journey. They provided financial support, consistently asked me how the program was progressing, and helped with little things that made a huge impact on my ability to endure in the program. I am forever indebted to them for all their sacrificial commitment, love, care, and support. Dear mom and dad, I love you both. I would not have been able to reach this milestone in my life without you.

Dr. Elizabeth Reilly, one of my first professors in the doctoral program and chairing member of my dissertation, made my entire LMU experience a thoroughly enjoyable and rewarding one. Her words “do not suffer in silence” in our first semester of the program gave me a sense of calm and quiet confidence that I was going to succeed in the program. That sentiment made it lucidly clear that cohort ‘16 was surrounded by caring and supportive professors like Dr. Reilly who would make certain to provide students the tools and support to succeed. I have mentioned this to her on numerous occasions, but I wanted to pronounce it once again to truly give it the weight it merits. Those simple, yet emphatic words meant so much.

When I started considering chair professors for my dissertation, it honestly was a very short process because Dr. Reilly immediately stood out in my mind. I was lucky enough for her to accept my request and from that day forward, she provided me with endless support, whether it was holding Zoom sessions with me, having conversation over the phone, and responding to my thousands of text questions and emails. Dr. Reilly, thank you so much for all your support. I could not have accomplished this feat without you.

Dr. Philip Molebash was introduced to me by Dr. Reilly, and I am very appreciative of this. From our first Zoom session together, I knew that Dr. Molebash would be a wonderful member of my dissertation team. He was gracious to accept my request and I am grateful for his time and efforts. His encouragement, especially in the final stages of the dissertation process, and overall feedback on improving my document were invaluable. Furthermore, his personable demeanor enhanced my confidence in my ability to complete the dissertation. Although I never mentioned this to you Dr. Molebash, but please know this was significant during my writing process. Thank you so much!

Dr. Peter Rich is an associate professor of instructional psychology and technology at Brigham Young University and authored many pieces of literature I read. Dr. Reilly and I thought it was worth a shot to ask if he would like to be a member of my dissertation committee. I contacted him via email, and we met thereafter via Zoom. With the conclusion of our session, he agreed to be another member of my dissertation team. It was humbling yet exciting to have a professor of his stature in the field of my dissertation's focus accept my request. Dr. Rich, I truly appreciate your guiding support during my doctoral journey. My dissertation's final product would simply not have been possible without you. Thank you!

Dr. Porsche Ruengvirayudh came to my rescue in the last few months of my doctoral journey when I needed to learn how to use SPSS and conduct various statistical analyses. She guided me through this arduous point patiently and with care. I cannot even guess how many questions I asked her, but she quickly responded every single time. I truly cannot have completed my dissertation without her support. Dr. Ruengvirayudh, please accept my most sincere gratitude.

Dr. Anthony Galla, Deputy Superintendent of the Department of Catholic Schools in the Archdiocese of Los Angeles, played a significant role in my success as well. I still recall my conversation with him as I applied to LMU’s doctoral program. He offered words of encouragement and advice that truly assisted me. Furthermore, during my doctoral journey, we had quite a few conversations, email exchanges, and Zoom sessions geared toward conducting research within the ADLA. Not only did Dr. Galla guide me through the process to request formal approval to conduct my study within the ADLA, but he approved this request and subsequently assisted me in identifying ways to gather data. I am humbled that he cared about my study and that he allocated time to support me. Dr. Galla, I truly appreciate your guidance.

I would also like to thank my colleague and friend Katy Monahan Huntley. I turned to her when various assignments required I receive input from a colleague. She was generous enough to fit me in her busy schedule, and I cannot thank her enough for it. Those assignments were ones that I had never done before; however, with Katy’s help, they became seamless and easy to complete. Katy, thank you for your assistance, encouragement, and support.

To Father Jim Bevacqua, our pastor at Holy Family Grade School, and Dr. Fidela Suelto, our principal at Holy Family Grade School, I am blessed to have amazing leaders such as yourselves. You have been reassuring from the very moment that you learned I would take part in this program. Your kind and reassuring words and prayers provided me with the confidence needed to continue to travail through the program. Please accept my sincerest gratitude.

One of the best things that came out of my time at LMU was befriending Antonio Felix, a fellow member of cohort ‘16. I am simply fortunate to have met him and I can unequivocally state he is one of the kindest, humblest, and most intelligent people I have ever come across. We conversed often about the program and the challenges that we faced. I appreciated Antonio’s continued support and guidance. Dear Antonio, thank you! We did it, brother!

Finally, to my wife Carla and our two sons, Nicholas and Matthew, I love you more than words can adequately convey. You inspire me every day to be a better husband and father, and I could not have asked for a more loving and supportive family I could not have accomplished this without you, and I hope you are as proud of me as I am of you. Thank you!

DEDICATION

I am blessed and privileged to be married to my wife, Carla. I imagined the rest of my life with her almost immediately after we met, and I consider myself extremely lucky to share my life with her. Words cannot truly encapsulate what she means to me; she is my best friend, my partner in life, my everything. We have experienced many cherished moments and challenges with each other, and I have grown to love her even more with each passing day. I treasure each day with her, whether it is something simple like our dinners together at home with our two sons, or our small getaways to Del Mar or San Diego, to all the momentous and small moments we have experienced together thus far. Earning a doctoral degree is yet another significant moment in our lives and I could not imagine sharing it with anyone else.

Carla is intelligent, strong, and loving and she has helped make up for my weaknesses and imperfections; she completes me. Her steadfast support was crucial in my doctoral journey and I cannot have accomplished this without her. I dedicate this achievement to my wife, Carla. I hope I have made her proud and I promise to continue loving and supporting her until my last day on Earth. To my dearest Carla, I love you more than words can possibly express. Thank you for all your love, care, and support. I share this achievement with you.

I also dedicate this accomplishment to my two dear sons, Nicholas and Matthew. After pondering whether to enter a doctoral program for a few years, I decided to begin this journey soon after our second son Matthew was born. They have inspired me during this trek and although they are a bit young to truly understand the meaning of a doctoral program, I hope I nevertheless made them proud. I also hope that this journey serves as an example for them to know that they can accomplish anything that they desire so long as they exhibit persistence. I am grateful to have them as my sons and I love them endlessly.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

ABSTRACT

Perceptions of Coding Instruction in K–12 Archdiocese of Los Angeles Catholic Schools

by

Krikor Kiladjian

Traditional pedagogy offers students opportunities to enhance various skills and acquire content knowledge; however, additional steps can be taken to enhance student achievement, prepare them for future occupations, and bridge the divide in access to technology. A curriculum that integrates coding instruction affords students the opportunity to augment their collaboration, communication, creative thinking, and problem-solving skills. This is especially crucial for traditionally marginalized populations who have experienced inequitable access to technology. Nevertheless, coding is not integrated in schools in different domains, including Catholic institutions in the Archdiocese of Los Angeles (ADLA).

This dissertation used a descriptive and inferential quantitative methodology to survey K–12 Catholic school teachers', administrators', and STEM directors' understanding of what coding entails, assess their perceptions of coding's potential to enrich student achievement, to prepare them for future occupations, and diversify STEM representation both in academics and in the workplace, and evaluate the potential link between educator epistemology and pedagogy with the penchant to incorporate coding instruction and the constructionist framework in the classroom.

The largest diocese of the country, the ADLA, was the sole focus of this study and the data demonstrated participants have a relatively limited understanding of what coding entails, but

they do believe it results in various benefits for students. Nevertheless, their epistemology and pedagogy are not ripe for constructionism to take hold in the classroom to facilitate coding.

CHAPTER 1

BACKGROUND OF STUDY

In an ever-evolving 21st century professional landscape, educators have the obligation to effectively prepare students to not only succeed in the classroom, but to also develop the skills needed to thrive in future fields of employment. Frey and Osborne's (2013) research asserted 47% of more than 700 current occupations are vulnerable to being digitized or computerized. Further, Casserly (2012) stated various positions (e.g., mobile application developer, digital strategist) have been developed in the past 10 years. Employers are not only seeking individuals who can follow instructions, but, more importantly, they are pursuing applicants who can solve novel problems collaboratively and creatively and deconstruct complex information (Darling-Hammond et al., 2017). Consequently, it is vital for students to be provided with the requisite foundation to respond effectually to these trends, and one means to do so is by integrating coding instruction in kindergarten to 12th-grade (K–12) classrooms. The purpose of this study was to gain insight into K–12 Archdiocese of Los Angeles (ADLA) teachers', principals', and science, technology, engineering, and mathematics (STEM) directors'/technology coordinators' understanding of what coding entails, evaluate their perceptions regarding coding's impact on academic achievement, occupational preparedness, and ability to enhance STEM access for traditionally marginalized groups, and whether participants' epistemology and pedagogy inclines them to integrate the tenets of constructionism and coding instruction.

Coding can be defined as the language that programmers use to instruct steps for a computer or application to take (CODE.org, n.d.-a). Alexiou-Ray et al. (2018) supported this description as they specified coding to be a set of instructions prescribed to a computer to fulfill a task. Alexiou-Ray et al. referenced the Berkman Klein Center's assertion that coding is a new literacy and that digital technologies are used to design and create in an embryonic society. Adams and Mowers (2013) even suggested coding is a global language that is used more routinely than languages such as English and Spanish. Wong et al. (2015) asserted coding develops essential competencies within students known as the "4Cs": critical thinking and problem-solving, communication, collaboration, and creativity/innovation. Additionally, according to Wing (2006), computational thinking (CT) is "interpreting code as data and data as code" (p. 33) and enables students to problem solve, design systems, and understand human behavior. In connection, Gove (2014) championed the reformulation of curriculum that integrates coding because programming allows students to solve computational problems and evaluate computational abstractions that represent real-world problems. Such skills prepare students for both present and forthcoming jobs (Gove, 2014). The notion of CT is taken a step further with the idea of computational participation by way of coding as students share their formulated products in a networked society with other likeminded creators (Burke et al., 2016). Falloon (2016) supported these assertions, stating skills, including cognition, collaboration, and self-management, are developed because of coding and are of significance for 21st century education. These points are bolstered with the following:

> To thrive in today's innovation-driven economy, workers need a different mix of skills than in the past. In addition to foundational skills like literacy and numeracy, they need competencies like collaboration, creativity, and problem-solving, and character qualities like persistence, curiosity, and initiative. (World Economic Forum, 2015, p. 2)

Burke et al. (2016) even declared that, much like how reading distinguished between those who were literate and illiterate, coding essentially serves the same purpose in our modern society.

In response to these growing demands in the workplace, in 2016, U.S. President Barack Obama instituted an initiative called Computer Science for All, which was aimed to expose students to computer science (CS) and computational thinking (CT) from kindergarten to high school so students become architects of technology, not just consumers (Price & Price-Mohr, 2018). This program was also in harmony with a survey demonstrating parents' general perception that CS was of equal relevance to other disciplines (Google, 2016). Vegas and Fowler (2020) posited coding is currently instructed in CS courses, which, according to Alexiou-Ray et al. (2018), can be integrated in the curriculum easily. Furthermore, "coding is now recognized as a critical literacy skill in the CS field" (Alexiou-Ray et al., 2018, p. 3). However, the resources afforded to coding instruction in the classroom is lacking, which manifests in a dearth of qualified candidates to fulfill technology-related positions. In fact, President Obama's initiative attempted to remedy the problem that 600,000 well-compensated, technology-related positions in industries such as transportation, healthcare, education, and financial services remained unfulfilled in 2015 (Smith, 2016), which corresponded to the general priority afforded to STEM learning in U.S. classrooms in the 21st century (Grover & Pea, 2013). Furthermore, according to the U.S. Bureau of Labor Statistics (2020), the median income for CS occupations in 2018 was $88,240, which was $48,000 more than the average of all other occupations in the United States. This was corroborated by Alexis Harrigan, director of state government affairs for CODE.org, who stated computing positions in California are rewarded an average of $110,000 in salary, while the average salary for all other positions equates to $56,000 (Lambert, 2019).

The Obama Administration’s policy was an attempt to develop invaluable skills within students and counter the dearth of CS courses offered in K–12 schools. Indeed, approximately only one quarter of all schools across the country offer CS with coding (Smith, 2016), and thus, the lack of skilled individuals to fulfill the aforementioned positions is, in part, attributed to this deficiency in schools. According to a Google and Gallup poll conducted in 2016, fewer than half of all K–12 schools nationwide offered a computer course that included programming (K12 Computer Science, n.d.). However, an increased emphasis on coding instruction in CS courses for all schools will naturally augment student interest and assuredly lead to broader and more diverse participation in this field in higher education and as a career. Indeed, one pivotal rationale for instituting this program was to give all Americans the opportunity to have technology-related careers (Smith, 2016).

In line with President Obama’s initiative, scholars in the field have suggested coding instruction should include grade–level students. In fact, whereas programming courses were solely offered at the university level in the past, there is now a growing sentiment that such instruction should be expanded to include younger students (Fincher & Utting, 2010). Various countries, including Germany, Sweden, the United Kingdom, and Australia already incorporated its instruction in the curriculum for all children (King, 2015). In fact, in Australia, the new digital technology curriculum requires students to learn coding in Year 5 of schooling (King, 2015). The United States has, to a degree, followed this trend as well, led by a slight political shift to incorporate it in schools in different states. In fact, according to Code.org Advocacy Coalition (2019), 34 states developed CS standards, and 26 states funded CS professional development opportunities. Although such progress serves as steps toward unified implementation of CS and coding courses in the country, it is insufficient to meet the current and forthcoming demands of an evolving global economy.

In addition to providing a basis to meet the demands of future workplaces, coding instruction can mitigate the reality that female students and students from traditionally marginalized segments of the population experience inequitable access to STEM related courses and programs, such as coding. In fact, there has been a persistent gap in the representation of ethnic minorities and socioeconomically disadvantaged students in the STEM fields (Blustein et al., 2013). This is distressing, as such fields typically provide opportunities for financial independence and upward mobility (Blustein et al., 2013). Native American, Black, and Latino students, along with those from lower socioeconomic groups and rural communities, have disproportionate access to CS courses (K12 Computer Science, n.d.). Additionally, statistics related to CS in higher education mirror the inequitable access to its instruction in K–12 classrooms. For instance, according to the National Center for Education Statistics (NCES, 2018), 19% of students graduating with a bachelor's degree in CS are women. Moreover, among all students who graduated with a STEM–related degree in academic year 2016–2017, 12.7% were Hispanic and 8.5% were Black (NCES, 2018). These figures undermine the purpose of technology according to the National Education Technology Plan (NETP), which should combat social exclusion and enhance access to education, educational technology, resources, and information (Roumell & Salajan, 2016). As one form of digital literacy, coding has the potential to animate learning because students directly and collaboratively create a product, such as a functioning robot. Additionally, providing opportunities for computational participation in the form of coding or programming-as-writing to marginalized groups strengthens their sense of empowerment (Hagge, 2018), which can not only lead to their increased participation in academic spheres such as technology and science, but can also enhance their future involvement in and contribution to a myriad of industries whose positions require the possession of skills that are developed as a result of learning how to code.

Problem Statement

Current and future occupations in the 21st century require unique sets of skills, such as the ability to deconstruct complex information and problem solve collaboratively (Darling-Hammond et al., 2017). According to Burnett and Merchant (2015), as literacies evolve in our society, there must be the concomitant effort to provide multimodal resources to students and shift the traditional literacies emphasized in schools. The ability to code, which is not a traditional literacy, can develop unique skills within students. In fact, “coding skills are seen as a combination of problem-solving, logical thinking, computational thinking, and design skills” (Tuomi et al., 2018, p. 420). However, because it is absent from the national curriculums and specifically Catholic school curriculums, students are not exposed to one crucial means to enhance their academic achievement and develop the skills deemed significant in 21st century workplaces. Despite coding’s importance, as suggested in research and in policy including the CS framework, coding’s instruction is not a consistent part of school curriculum. Its instruction is generally a localized decision and thus depends on the preferences and available resources of the school. Regrettably, this exacerbates the well-documented gap in access to STEM opportunities in schools and to careers associated with those disciplines for female students and those from marginalized populations.

Students from marginalized groups do not receive the requisite exposure to be engaged in and pursue advanced degrees in STEM careers. Inequities in urban schools is rooted upon long-enduring structural dynamics based upon slavery, racism, and segregation (Avendano et al., 2018). Unfortunately, educational injustices persisted; according to Sparks and Pole (2019), a disproportionate number of ethnic and racial minority students who entered STEM fields continued. The effects of this inequality include the reality that in 2015, only 13% of the Advanced Placement (AP) Computer Science examinees were African American or Latino students (Smith, 2016). Additionally, according to Allison Scott, chief research officer at the Kapor Center, although Black, Latino, and Native American students comprise 60% of the high school population in California, they account for only 16% of enrollees in AP CS courses (Lambert, 2019). Furthermore, technology firms reported only 3% of their employees are Black (Smith, 2016). One basis for the general disparity in access lies in the White, male-dominated STEM power structures, which may dissuade students of color from entering the fields as a potential career avenue (Sparks & Pole, 2019). Although such realities paint a bleak picture, there is the prospect to reverse this course. Affording students from marginalized groups access to code from an early age in school will inevitably engender at least a modicum of interest in the discipline and subsequently result in the pursuit of higher education and careers related to it. Consequently, coding instruction has the potential to contract the divide in STEM diversity.

Similar to minority underrepresentation in the STEM fields, there is a pervasive gender gap as well (Master & Meltzoff, 2017). Sparks and Pole (2019) cited research indicating female underrepresentation in the STEM fields as well. According to the European Union and the Organization for Economic Cooperation and Development, as referenced to by Master and

Meltzoff (2017), compared to men, women are less likely to earn STEM degrees and hold STEM careers. Young students' early academic interest and motivation in STEM fields is vital to their future careers; however, the eventual disparities in advanced degrees and careers are rooted partly in stereotypes (Master & Meltzoff, 2017). Boys are typically associated with STEM, are perceived to be more adept at problem solving, and are more culturally accepted to engage in subjects such as math and technology (Master & Meltzoff, 2017). These stereotypes, to a large degree, prevent females from pursuing STEM education and careers (Avendano et al., 2018). In fact, although STEM careers are becoming more prominent in the United States, women consist of a small proportion of the workforce (Avendano et al., 2018). There is no basis for insensitive misconceptions as ability in CS education or coding and such professions do not derive from innate aptitudes; rather, it simply centers upon access to content (Vegas & Fowler, 2020). As another microcosmic example of a lack of diversity in STEM fields, among students who took the AP Computer Science exam in 2015, only 22% of examinees were women (Smith, 2016). Moreover, such statistics are reflected in the gender makeup of employees in technology firms as less than one third are women (Smith, 2016). Furthermore, women's representation in engineering accounted for only 19% and 18% in CS in 2012 (Master & Meltzoff, 2017).

Because CS and technology will continue to be prevalent parts of society, this gender gap must be eradicated. One means to accomplish this is with feeder courses; foregoing even one can have an adverse impact on a student's motivation and willingness to enter a STEM field (Master & Meltzoff, 2017). Girls may simply be less involved in STEM because they have fewer opportunities that can ignite their interest (Master & Meltzoff, 2017). Based on these lines of reasoning, offering coding instruction will inevitably result in at least a measure of increased interest in female participation in STEM oriented academic and career fields.

There should not only be equitable access to academic, but career opportunities in the STEM fields as well. Roumell and Salajan (2016) deemed technology an access point to education and if these words are heeded by educational leaders by their decision to implement coding instruction in the curriculum, it can also serve as a springboard for a more diversified population in the STEM occupational fields. Specifically, coding instruction can remedy the aforementioned ills by affording students of all ethnic backgrounds and female students the opportunity to learn coding in the classroom, which can engender interest and a pursuit of higher education and careers related to coding. In turn, this will diversify the pool of people who hold coding related positions, which will thereby enhance the representation of STEM related occupations. There is an increasing perception in the U.S. public that STEM subjects and coding instruction are crucial for students' prospects at success (PR Newswire, 2015). States have and continue to take gradual steps to infuse CS in the classrooms (Code.org Advocacy Coalition, 2019), yet such innovative thinking is still not unanimous among educational leaders and governmental policymakers in the country. Additional effort is needed to shift perceptions regarding CS instruction and coding, and it is my belief that this dissertation, which was based on underscoring various stakeholders' perceptions in ADLA schools, will yield further testimony that transformation is needed in our educational curriculum. Only with a shift in beliefs can true change in action take root. After all, Owens and Valesky (2015) stated, "Actions—that is, behaviors—flow from the values and beliefs that we embrace" (p. 24). If, as educational leaders, we want to engender and maintain not only a pedagogical system providing all students additional opportunities to succeed in the future but create a more socially just and equitable educational environment for marginalized segments of students, then long-held assumptions and beliefs must be confronted and forward-thinking energies must instead commence. According to Owens and Valesky (2015):

> If we want to make a difference in the organization we call school, it is first necessary to carefully make our basic assumptions manifest and consider how logical the connections are between those assumptions, our publicly espoused values and beliefs, and the organizational behavior that we use in professional practice. (p. 25)

As coding and CS gain greater public attention and figure more prominently in discussions among educational leaders, district, diocesan, and governmental policy will ensue. Consequently, assessing people's perceptions and enhancing their knowledge and curiosity about coding and CS will serve to accomplish this.

First, based on the literature I examined, I did not come across any demonstrating a curriculum inclusive of coding instruction in Catholic schools under the umbrella of the ADLA. Second, as a Catholic school teacher in the ADLA for the last 12 years who has participated in professional development sessions, coding has not been a topic of discussion; generally, the over 250 ADLA schools' curriculum does not integrate consistent coding instruction. As a Catholic school educator, I pride myself on facilitating students' spiritual and moral development, but I also correspondingly underscore the importance of academic rigor and preparation for forthcoming challenges in secondary school, higher education, and the workplace. By integrating coding instruction into the Catholic school curriculum, we, as a school system, will better prepare our students for these challenges. My study's purpose was to bridge the gap in relevant literature on this issue in ADLA schools and to integrate coding in ADLA's curriculum. By fulfilling these dual purposes, Catholic schools will have a better grasp of what coding entails, recognize its consequent benefits, and have a blueprint on how to mold their pedagogy to meet this end. An earnest commitment to this end will engender a new generation of Archdiocesan students equipped with 21st century skills that will enhance their academic achievement, augment their prospects for employment positions in a wide spectrum of fields, and thus diversify the STEM field.

Research Questions

1. What is K–12 ADLA Catholic school teachers', administrators', and STEM directors'/technology coordinators' understanding of coding?
2. What is their perception on its instruction's impact on students' academic achievement, occupational preparedness, and on its ability to enhance STEM representation in school and in the workforce?
3. How do K–12 ADLA Catholic teachers', administrators', and STEM directors'/coordinators' epistemology and pedagogy impact their inclination to integrate the tenets of constructionism and coding?

In addition to assessing the perceptions of the entire pool of participants, the data will be organized and evaluated based on various subgroupings, which will be explicated in Chapter 3. The following hypotheses apply to each research question:

1. Null Hypothesis: There will be no statistically significant difference in means of perceptions between subgroups.
2. Alternate Hypothesis: There will be a statistically significant difference in means of perceptions between subgroups.

Purpose

The aim of this research was three-fold: first, identify what teachers, administrators, and STEM directors/technology coordinators understand about coding, secondly, assess their perceptions on coding's impact on student achievement in the classroom, occupational preparedness, and its potential to bridge the divide in STEM representation both in schools and in the workforce, and third, gauge the linkage between educator epistemology and pedagogy with the integration of constructionism and coding. The second point was vital, as coding can be a means to promote social justice by providing greater opportunities to students who have traditionally been marginalized from the STEM fields. This study focused on Catholic schools in the ADLA, which is the largest Catholic diocese in the nation. There was increased prominence accorded to STEM disciplines, including CS and coding instruction, by educational and legislative leaders. Additionally, there is a growing sentiment among the U.S. public that CS and coding instruction are essential for future success. Consequently, I intended to amplify this general sentiment and thus begin to take steps toward a uniform shift in curriculum that embraces coding in Catholic schools across the nation.

Significance

This study will intensify the call to incorporate coding instruction in curriculums; however, it will focus on the ADLA. The ADLA is the largest diocese in the United States, and its Department of Catholic Schools superintends 265 schools with thousands of students; however, the curriculum in these schools lacks consistent coding instruction. As a Catholic school teacher for the past 12 years in the ADLA, it is my humble hope that my study will ideally not only assist in addressing the current gap in literature regarding Catholic school curriculums and coding education, but firmly encourage its integration by highlighting its benefits and a pedagogical foundation with which to do so.

This study moreover has the potential to spread and penetrate all curriculums across the country by informing educational leaders and policymakers at the local, state, and federal government levels about the benefits of and the means to employ coding instruction. Different countries, including Germany, the United Kingdom, and Australia, have already incorporated coding in their curriculum (King, 2015), but the U.S. national curriculum is still devoid of its consistent instruction. Coding provides students the opportunity to carve a space for themselves in our evolving digital society (Burke et al., 2016). According to Falloon (2016), integrating coding into primary school curricula allows students to "exercise a range of general, computational, and higher-order thinking skills" (p. 591). As such, a transformative shift to implement coding will provide the opportunity to develop such competencies in all students, which can enhance their prospects for future academic and professional success. Even more profoundly, such opportunities will unlock academic and occupational doors for students who have been traditionally marginalized, such as female, Black, and Latino students from fields such as CS, technology, mathematics, and engineering. King (2015) buttressed this point, positing female students are underrepresented in CS, yet, according to research from the University of Ohio, female interest in STEM fields is robust at earlier ages but dissipates later in their academic life. Coding apps, however, can engage them and simply show it can be exciting and fun (King, 2015). Falloon (2016) emphatically stated, "[Coding] is very important given the role of school curricula in helping prepare *all* young people for future citizenship, not just those wishing to pursue technology-related careers" (p. 591). Falloon emphasized the word "all" because it is imperative that students, regardless of gender, race, or creed, or any other personally identifiable characteristic be given equitable opportunities to succeed. Additionally, providing these opportunities for digital participation to marginalized groups strengthens their sense of empowerment (Hagge, 2018), which can translate to success in employment fields related to technology and ultimately, bolster and diversify the current STEM pipeline.

Theoretical Framework

Constructionism was the focusing lens to conduct my study. This paradigm provided a firm rationale to use technology in the learning and teaching processes. Here, a summation of constructionism is provided; however, a comprehensive explanation will follow in the beginning of Chapter 2, as well as a discussion of constructivism, which is the framework that constructionism is founded upon.

Constructionism

Constructionism is a learning and teaching framework developed and championed by Papert (1999), a mathematician and educator. This model asserts students of all ages and social backgrounds can program if they possess the requisite resources and are afforded the opportunity (Papert, 1999). Consequently, students can achieve more than what is perceived to be capable (Papert, 1999). Access to technology is important, but developing an intellectual culture allowing students to interact with powerful ideas and construct individual projects is paramount (Papert, 1999). According to this theory, technology should not simply serve as an informational medium, which schools often defer to; rather, technology should be used as a constructional medium to enhance learning (Papert, 1999). Indeed, the use of computers has the potential to evolve the learning process as the educator impetus to transfer knowledge to students shifts to allowing them to produce it (Papert & Harel, 1991). Indeed, this pedagogical revolution can inculcate the various coveted skills within students during an everchanging 21st digital century.

Papert's constructionist philosophy was geared to enhance student learning by way of creating tangible products or artefacts (Papert & Harel, 1991). I contend his vision can be championed using any programming or coding language, such as Scratch (https://scratch.mit.edu), ScratchJr (https://www.scratchjr.org), or Python, so long as students are afforded the platform to collaborate, construct, and thereby participate in a different modality of learning. Noss and Clayson (2015) stated, "Constructionism symbolized a way of thinking about learning, a metaphor for the ways that human beings come to learn most effectively: building a model, reflecting on it, debugging and sharing" (p. 285). Furthermore, Beynon and Roe (2004) described constructionism as active learning and ultimately, what is of greatest significance is not what is assembled but the reasoning processes involved in formulating the artefact. This process produces multiple benefits including the capability for learners to pursue their interests, visualize the results of their work, and communicate their learnings and findings (Beynon & Roe, 2004).

Stager (2010) accentuated a powerful prescription of constructionism as well: teachers collaborate with students and empower them to formulate their knowledge by allowing them to construct an artefact. Such an educational setting where knowledge is constructed through collaboration requires significant changes in traditional curriculum and pedagogy (Stager, 2005). This position mirrors mine, especially regarding curriculum, because I maintain our curriculum should integrate a set coding course or embed it within the instruction of other courses to enhance students' learning opportunities, development of pivotal skills, and preparation for current and forthcoming occupational demands. Moreover, Papert (1999) even suggested students who are not given opportunities to develop the skills requisite to create and construct will lack independence in society. A logical extension of Papert's recommendation is that students who participate in knowledge construction via coding will be more aptly suited to meet the evolving demands of a wide range of occupations, even those not necessarily centered upon technology.

Methodology

A quantitative descriptive and inferential study was employed to gather empirical data on K–12 ADLA Catholic school teachers', principals', and STEM directors'/technology coordinators' understanding of coding, their perceptions of coding instruction's impact on academic achievement, occupational preparedness, and diversifying the STEM field, and to gauge whether their epistemology and pedagogy impact the proclivity to integrate constructionism and coding in the classroom. The study took the form of a survey, which was formulated based on adopted items from previous literature and novel questions I developed.

Quantitative Research

The purpose of survey research is to test a hypothesis by gathering sufficient data or to answer a question about people's preferences and attitudes on an issue or problem (Mills & Gay, 2019). My study focused on the latter, as I obtained data on people's opinions regarding whether coding instruction is needed in the curriculum; to what degree, if any, it enhances students' prospects for academic achievement and occupational preparedness; whether it promotes social justice ends; and whether an educator's pedagogy and epistemology impact their inclination to integrate constructionism and coding. A cross-sectional survey design, which obtains data at a particular point in time (Leavy, 2017), was an effective and efficient means to obtain a wealth of information. The data were thereafter used to respond to my research questions. Preexisting surveys (Leavy, 2017) corresponding to the points of my inquiry were incorporated and tailored to a degree to suit my specific purposes. I did not allow my positionality as a STEM director and an educator who favors the integration of coding instruction in the curriculum to influence modification of the items. The questionnaire was comprised of fixed-choice, Likert-scale items (Leavy, 2017), which were subsequently analyzed statistically with measures including means, modes, standard deviations, and percentages. The findings are generalizable to the larger population of the ADLA's teachers, administrators, and STEM coordinators.

Limitations/Delimitations/Assumptions

A limitation of the study was time. Due to the parameters of Loyola Marymount University's (LMU) doctoral program in educational leadership for social justice, the entire data gathering process was limited to approximately 5 months, commencing in June 2021 and concluding in October 2021. Although a lengthier period of data collection would have yielded additional data, I remained confident the responses in the abbreviated timeframe would be sufficient to answer the study's research questions. A second limitation was the number of responses informing my study. The daily schedule of educators and other personnel in a school environment is invariably hectic, thus adversely impacting the response rate. Third, given the lack of a curriculum that infuses coding instruction in the Catholic school curriculum, I had to assess shareholders' perceptions of the implementation of coding instruction. I contend perceptions and beliefs serve as powerful bases to effect change; however, I am unable to further substantiate my claims by demonstrating the impact of coding instruction on substantive student achievement in the near term, their prospects for positions related to coding, or whether it diminished the gap in access and opportunity for marginalized groups. Not only would a

Catholic school need to have a consistent coding program, but a longitudinal study would be necessary to evaluate these variables. A future study of mine or another researcher may inquire into a longer term study regarding these questions. The fourth limitation regarded the creation of constructs. I did not validate them empirically using factor analysis; however, based on the literature and theories, I was confident the selected items harmonized to create each construct. A fifth limitation was the global COVID-19 pandemic, which began in the early months of 2020. Generally, survey response rates have declined, but with a pandemic further exacerbating challenges in the school environment, educators and administrators were less inclined to participate. Some principals explicitly stated the situation at the time of my request prevented their participation.

A delimitation was my sole focus on K–12 Catholic schools in the ADLA. As an educator for 12 years at a Catholic school in the ADLA, I felt impelled to begin formulating change in this context. Although my ultimate objective is to engender change in all curriculums, I decided to begin with a study on the ADLA. This delimitation precludes the study's generalizability to schools in other domains, such as public and charter, yet I hope that this study propels future researchers to conduct similar studies in those academic spheres. My focus on K–12 schools stemmed from the prevailing literature's emphasis on the consequence of offering coding instruction at an early age. It additionally stems from literature asserting middle school years are formative in a student's future educational and professional aspirations. A second delimitation is my emphasis on teachers, administrators, and STEM directors/technology coordinators. My study was more interested in their understandings and perceptions related to coding. Parents and students were excluded to ensure the study was focused and manageable.

Nevertheless, future studies on coding can be conducted using those populations. Third, I discussed the importance of CS instruction, but I made a conscious choice to focus on coding instruction, which is a narrow part of CS.

My assumption is that coding instruction in the K–12 classrooms will enhance student interest and spur their gravitation toward higher education and professional fields related to it. Additionally, I am of the assumption that a curriculum that embraces coding instruction better prepares students for the occupational shifts in modern global economics by specifically positioning them to not only obtain positions related to coding, but ones that prioritize the skills that are developed as a result of coding in a wide variety of industries.

Conclusion

Shifts in global economics have resulted in not only evolved demands in workspaces, but in the creation of novel positions. Positions in instructional technology (IT) and coding continue to expand and skills such as critical thought, creativity, collaboration, and algorithmic thinking are highly sought. However, there is a current paucity of individuals who can fulfill this demand, yet an emphasis on coding instruction in K–12 classrooms possesses the potential to remedy this reality. Additionally, literature suggests coding can enhance student achievement in the classroom and diversify the current population of those who hold STEM related positions. This latter point is especially noteworthy as segments of the population including females, African Americans, and Latin individuals are disproportionately represented in STEM related positions, including coding/programming. However, if all students from kindergarten are exposed to coding instruction, it will create more opportunities for students and most assuredly germinate into some pursuing related academic and vocational ends. To accomplish this, coding must be integrated into the academic curriculum and tangible steps must be taken to prepare educators to incorporate it. Constructionism, which is a theoretical framework that champions the use of technology for creating novel products or artefacts, can serve as the vehicle to effectuate this objective in the classroom. As a Catholic school teacher for the past 12 years, I humbly, yet firmly believe coding's incorporation in the curriculum will lead to substantial benefits for students both in the immediate term as well as in the future. My study's focus on Catholic schools in the ADLA will assess whether others hold this perception. I am hopeful that the results will serve as a springboard for change in Catholic school curriculums.

Chapter 1 presented the structure of this dissertation as it consisted of crucial elements including its purpose, significance, research questions, and methodology. Chapter 2 will delve into the prevalent literature on coding instruction, including a discussion of significant concepts such as CT, the various benefits that stem from coding instruction, and the potential means of implementation, including challenges to effectuate it. Chapter 3 will explicate the methodology that will base this study, which will primarily include a discussion of quantitative design, the study's participants, and the survey instrumentation. Chapter 4 will highlight the analysis of the data, including detailed connections between various points of interest and overarching observations. The study will conclude with the Chapter 5's discussion of the findings, its implications, and recommendations for action and future research.

CHAPTER 2

LITERATURE REVIEW

This chapter will focus on two crucial elements of the dissertation. First, an exhaustive analysis of constructionism will demonstrate the foundational precepts of the framework and provide the rationale for its application, including specific examples of its utility in the classroom. The literature review will follow, and it will carry greater relevance perched on the backdrop of constructionism. The literature will underscore concepts pertinent to coding, including CS and CT, it will detail the benefits associated with coding's instruction, highlighted by heightened academic achievement, enhanced occupational preparedness, and ability to contract the incongruence of STEM academic and occupational representation, and expound upon the potential means of implementation and the challenges that may arise from it.

Theoretical Framework

Constructionism is one vehicle through which coding can be championed in the classroom. Coding is at the very essence of what Papert, the architect of constructionism, believed—use technology beyond its instructional ability and provide students with the tools necessary to create innovative products (Papert & Harel, 1991) that simultaneously engender the development of skills such as critical thought, algorithmic thinking, communication, and collaboration. Coding allows students to collaborate and critically think to produce creative artefacts such as multi-functional robots. Coding's instruction can also activate greater student engagement and allow them to become the drivers of their educational process, which are additional critical components of constructionism. Constructivism, which is an alternate framework to constructionism, suggested learning is building knowledge; however, constructionism added to this idea by asserting learning especially takes place when the learner is engaged in constructing a tangible product (Papert & Harel, 1991). The following portion of Chapter 2 will begin with a brief summation of constructivism and proceed with a focus on constructionism.

Constructivism

Constructivism, an educational model proposed by Piaget, asserts knowledge is constructed by the learner instead of it simply being transferred by the teacher (Rob & Rob, 2018). Alanazi (2016) wrote, "According to Piaget, the knowledge people interact with is added to schemas of prior knowledge wherein learners construct knowledge" (p. 2). Rob and Rob (2018) further asserted, "Piaget's constructivism is a theory of knowledge that argues that humans generate knowledge and meaning from an interaction between their experiences and ideas" (p. 275). Stager (2001) furthered this point, stating, "Each individual must reconstruct knowledge and this learning process happens within a material environment, a culture and a supportive community of practice" (p. 5). Consequently, concepts are understood and perceived differently depending on the learner's prior experiences (Alanazi, 2016). Constructivism additionally maintains knowledge is not fixed nor is it external to the learner; rather, attainment of knowledge is based on novel experiences and ideas (Rob & Rob, 2018). Piaget strongly contended students are not empty vessels to be filled by knowledge emanating from educators (Beisser & Gillespie, 2003). Thus, constructivism centers the responsibility of students in the learning and teaching processes rather than limit them to a secondary role. Sentance and Csizmadia (2016) stated, "[Constructivism] implies a need for authentic and meaningful experiences to support learning based on prior experiences and models of the world" (p. 490).

Sentance's and Csizmadia's (2016) study identified teachers' belief that a constructivist pedagogy aids in the instruction of programming. Indeed, a constructivist pedagogy is deemed to be more sophisticated relative to a traditional form of teaching because the latter places teachers in a central position in the classroom and students take on a passive role (Chan & Elliot, 2004).

Educators' perception that constructivism lends itself to instruction and student comprehension and application of coding yields optimism for constructionism's potential to not only support it as well, but because this paradigm emphasizes the use of technology, it offers even greater potential.

Constructionism

Papert was not only a mathematician and educator who developed the Logo computer language (Beisser & Gillespie, 2003) but also was the pioneering founder of constructionism, a theoretical paradigm inspired by constructivism (Rob & Rob, 2018). Papert served as professor of learning research and director of the Epistemology and Learning Group at Massachusetts Institute of Technology, and his Logo language stimulated the LEGO company's creation of the Mindstorms robotics kit (Beisser & Gillespie, 2003). Papert was a protégé of Piaget's and the latter's constructivism laid the foundation for constructionism (Badilla-Saxe, 2010). Constructionism not only foundationally entails learning by making (as cited in Papert & Harel, 1991) using technology in an academic environment, but it also emphasizes the significance of sharing the product with others to maximize the benefits of this process (Badilla-Saxe, 2010; Rob & Rob, 2018). Harel and Papert (1990) posited, "Our vision focuses on using technology to support excellence in teaching, in learning, and in thinking with computers—technology as a medium for expression” (p. 4). Stager (2001) offered a compelling perspective espousing the importance of the computer as the ultimate tool for learning:

> For learners, the computer provides an unrivaled intellectual laboratory and vehicle for self-expression. The computer becomes the workspace within which students can, for example program video games, construct simulations, perform calculations, store their journals, publish newsletters, correspond with experts, edit video, produce animated films, learn to fly an airplane and much more. (p. 6)

Moreover, a fundamental component of constructionism is active student engagement, which leads to the creation of tangible products. Thus, a significant divergence between constructionism and constructivism is the former’s insistence on constructing tangible objects, whereas the latter deems learning as a cognitive incidence (Alanazi, 2016). In fact, Papert was critical of constructivism’s lack of context to apply what is being learned (Stager, 2001). A constructionist setting, in contrast, upholds the importance of creation and it considers it as the very manifestation of learning (Alanazi, 2016).

Papert also underscored the importance of learning through the trial-and-error process (Stager, 2005). Students will inevitably make mistakes; however, this affords them the opportunity to modify their work. According to Rob and Rob (2018), “Thus, at the heart of constructionism lies the belief that learning occurs in the process of creating a product that can be shared with others. Many authors contend that the creation and sharing of a product occurs effectively in a collaborative learning environment” (p. 276). Such a platform is especially empowering for students because it yields at least a modicum of control to them; they become active and incentivized agents (Badilla-Saxe, 2010) during the knowledge acquisition and creation process. Alanazi (2016) buttressed these contentions, positing “knowledge becomes constructed where complex problems and real issues arise in the learning environments and, specifically, where learners are engaged and involved” (p. 5). Also, Rob and Rob (2018) stated:

> The classroom is no longer a place where the teacher acts as an expert to “pour knowledge” to “passive students.” In this approach, the students are asked to be actively involved in their own process of learning. The teacher acts as a facilitator who coaches, Mediates, prompts and helps students develop and assess their understanding, thereby promoting their learning in a higher level. (p. 279)

Teachers create an environment for students to succeed and guide them in achieving the lesson’s tasks; however, the students are collaboratively engaged in crafting the product, reflecting upon it, and sharing the result (Rob & Rob, 2018).

Constructionism also emphasizes the art of learning and the procedure involved in creating a product (Rob & Rob, 2018). According to Rob and Rob (2018), “Because of its emphasis on learning through collaboration, building a meaningful artifact, sharing the artifact, as well as the use of tools, media and context, we argue that constructionism provides a better learning opportunity than constructivism” (p. 278). This position is supported by constructionists who suggest “learning is active and superior to a pedagogy of learning by telling. They value a plurality of definitions, meanings and ways of knowing. Learning is highly personal and controlled by the learner” (Stager, 2001, p. 6). Badilla-Saxe (2010) endorsed the praise heaped on constructionism by averring, “In a constructionist environment, rich in meaningful experiences and where the students are allowed to take an active role in their own learning process, unforeseen knowledge will emerge” (p. 9). Indeed, the development of such knowledges, social skills, and habits of mind are the precise requisites to make lasting and crucial contribution to the 21st century (Stager, 2001)

Constructionism possesses several characteristics (Noss & Clayson, 2015). *Modelling*, by way of building, reflecting, and debugging, lays a powerful base of knowledge within students, and it contravenes traditional models of teacher pedagogy (Noss & Clayson, 2015). It allows students to be the drivers of their learning and academic experiences rather than the educator. *Accessibility* serves as the second characteristic in the constructionism agenda—technology is an omnipresent part of students' lives, and, consequently, they must be afforded the opportunity to comprehend how they function (Noss & Clayson, 2015). *Layering* gradually integrates science and mathematics concepts in the use of the software, which enhances students' problem-solving capacities (Noss & Clayson, 2015). *Tapping into youth culture*, which is a consistent challenge for educators, is the fourth characteristic of this framework (Noss & Clayson, 2015). Being in line with students' agendas (Noss & Clayson, 2015) and adopting a pedagogical perspective that illuminates the importance of and their connection to math, science, and other subjects will create a powerful incentive for students to engage in their learning process on a heightened level. Furthermore, students' opportunities to *represent* themselves by way of expression or communication makes learning or knowledge visible (Noss & Clayson, 2015). Finally, as the sixth characteristic, *collaboration* serves as a vital cog in the constructionist mechanism; students take individual and collective responsibility in the acquisition of content using technology (Noss & Clayson, 2015).

Constructionism at Play in the Classroom

Harel and Papert (1990) illuminated Noss and Clayson's (2015) characteristics of constructionism with fourth-grade students learning mathematics. Students were tasked with designing instructional software but were required to discuss something about fractions to their audience (Harel & Papert, 1990). Students, using Papert's Logo language, experienced palpable advancement in their knowledge of fractions (Harel & Papert, 1990). Pre- and post-testing evinced this achievement, as students who had scored slightly above 50% before the exposure to the activity had thereafter scored percentages in the 80s and 90s (Harel & Papert, 1990). Students' attitudes toward fractions also improved due to this activity, which allowed them to use their previous knowledge about fractions rather than follow set procedures regarding how to learn fractions (Harel & Papert, 1990). In fact, "what seemed to make fractions interesting to these students was that they could work with them in a context that mobilized creativity, personal knowledge, and a sense of doing something more important than just getting a correct answer" (Harel & Papert, 1990, p. 6). Harel and Papert stated, although questions can be raised about whether the achievement was solely attributable to Logo, it was their firm belief that it was because the software allowed students to connect concepts and their representations and modify them continuously. Harel and Papert (1990) also attributed this learning to communication that comes by way of constructionism, which was emphasized with the following:

> Logo facilitated this ongoing personal engagement and gradual change of knowledge; and at the same time, it also facilitates the sharing of the knowledge with other members of the design studio, and it allowed learners to continue and build upon their and others' ideas and comments very easily. Logo facilitated communications about the processes

> and acts of cognition and learning. (p. 25)

This unique classroom environment, which granted students greater autonomy and offered a different pedagogy, enhanced students' achievement in a content area in which they previously struggled. This demonstrates the potential of constructionism as a viable framework to facilitate student progress and learning through coding.

Stager (2001) provided yet another application of constructionism in an educational setting, highlighting the Constructionist Learning Laboratory (CLL) at the Main Youth Center (MYC), which is a state facility serving youth and young adults with criminal histories. Stager (2001) wrote:

> The CLL strives to create an environment in which young people are engaged constructively, but without the coercion so often associated with traditional curricula. The absence of a bell schedule, tests and artificially segregated subject-area classes allows students to make connections between disciplines. They can work in depth on personally meaningful projects without the disruptions associated with high-stakes testing or competition fostered by traditional schooling. Students have the time to make mistakes, redefine their goals and develop the technological fluency required for realizing their objectives. (p. 4)

CLL students used programmable Legos complete with sensors, gears, motors, and bricks to create various inventions (Stager, 2001). For instance, students built robotic arms, machines that play the xylophone, and gearboxes that can pull objects with significant weight (Stager, 2001). Students also were expected to communicate their design and construction processes and reflect on it (Stager, 2001).

The CLL inculcated a unique culture immersed in collaboration and a firm sense of community, which are crucial skills in life (Stager, 2001). Students have already developed skills in content areas including film making, drama, video game design, and electronics, which are then applied in their construction of physical projects (Stager, 2001). As such, “CLL students are active knowledge workers rather than passive recipients of content produced by others” (Stager, 2001, p. 5). This simple, yet powerful idea gives students the opportunity to reason critically, collaborate, and construct products that are a materialization of their learning and creativity. In fact, “Constructionists are concerned with the goal of re-empowering the powerful ideas learned by students by taking a step towards re-empowering the idea of learning by discovery” (Stager, 2001, p. 6). This form of pedagogy offers an alternate means to develop the very skills needed in modern society and it champions the needs of students by positioning them as the drivers of their learning.

As was noted with Stager’s (2005) CLL example, one potential means to animate constructionism in the classroom is through robotics (Stager, 2010). The LEGO Mindstorms set, for instance, is comprised of gears, sensors, and motors allowing students to enliven the robot they create from LEGO pieces with Papert’s Logo language (Beisser & Gillespie, 2003). Beisser and Gillespie (2003) detailed how first-year undergraduate students, who would be future educators, learned how to engage students using technology. They used programmable LEGOs and became well versed in using motors, gears, and sensors (Beisser & Gillespie, 2003). The undergraduate students learned how to think critically, inquire, seek peer assistance, problem solve, be persistent, and reflect in a classroom modeled upon constructionism (Beisser & Gillespie, 2003). Stager (2010) maintained robotics can be used as a conduit for self-expression, to teach science-based concepts such as force and friction, mathematical concepts such as fractions and variables, CS concepts including programming [coding] and debugging, and problem solving. Furthermore, Stager (2010) suggested:

> the role of the teacher is to create the productive context for learning, including material organization, scaffolding, consulting, collaboration and anticipating forthcoming needs of each student. Teaching is subordinate to learning and the teacher is available to seize teachable moments and collaborate with students, rather than direct activity. (p. 2)

Robotics upholds a major precept of constructionism, which is to facilitate students' proactive role in the learning process. One example Stager (2010) shared was a 15-year-old's reconstruction of Thomas Edison's phonograph. The student was incarcerated and had experienced difficulty with prior academic work; however, he was able to produce a robotic phonograph with no previous experience with such an activity (Stager, 2010). He used LEGO materials, programmed the phonograph, and developed key vocabulary to communicate what he created (Stager, 2010). Such testimony underscores Stager's (2010) position that "students served by a constructionist approach grow in ways beyond the typical goals of a robotics project. An at-risk student raises their personal educational standards when in a rich environment that places their talents, needs and expertise ahead of standard curricular requirements" (p. 6). This assertion is in harmony with Papert's vision of an academic environment devoid of a standard curriculum (Stager, 2010). Papert promoted a structure in which students learn mathematical, scientific, managerial, and project skills and concepts they will use (Stager, 2010). Stager (2010) encapsulated his assertions with the following:

> Children have a remarkable capacity for intensity. It is incumbent upon educators to leverage this gift in order to support children in exceeding their potential. Too often, our focus on achievement, assessment and curriculum coverage creates an artificial ceiling. Having the courage to behave as if learning is natural and children are competent liberates students and teachers to achieve in powerful ways and at a level previously unimaginable. (p. 9)

Stager's example highlights the potential of student achievement when constructionism and coding are infused in an academic setting.

Savage et al. (2003) provided yet another example of student progress when integrating constructionism and robotics in the classroom with their discussion of how the Lego Mindstorm Robotics System (LMRS) was used by students to create a programmable brick-based robot with sensors and motors. A personal computer (PC), through an infrared connection, provided instructions to the robot to engage with the environment using its sensors (Savage et al., 2003). Before students engaged with the LMRS, a discussion was held to introduce the robot and explain how to use its interface (Savage et al., 2003). Thereafter, students were divided into groups of 4, and each group was given a specific task to achieve with the robot. Upon completion of the activity, each group presented their work and reflected on the learning process (Savage et al., 2003). Ultimately, combining constructionism and robotics in the classroom promoted the development of higher order thinking skills within students (Savage et al., 2003).

The literature review that follows will highlight important concepts such as CT and provide a more in-depth explanation of what coding entails. It will additionally underscore the various benefits that result from coding's instruction, including enhanced student achievement, greater student preparation to meet the demands of technology and non-technology centered workspaces, and the potential to variegate the populations that typically hold STEM related positions. Lastly, the literature review will accentuate some potential means to incorporate coding instruction in the classroom and call attention to the challenges therein.

Literature Review

The traditional model of curriculum that incorporates disciplines such as language, science, social studies, and math provide a firm foundation for students to be prepared for and be successful in various occupations in the future; however, as the 21st century workplace continues to evolve, a specific set of skills are required to meet heightened demands. Coding instruction has the potential to inculcate such invaluable skills including critical thinking, creativity, collaboration, and communication. These skills will enhance student achievement in the classroom and prepare them for positions in numerous fields of occupation that are not only centered upon technology, but in sectors shifting toward digitization. Moreover, coding offers one avenue to bridge the persistent divide in STEM access and opportunity for female students and those from marginalized populations. Ideally, schools are to provide equitable opportunities for students to succeed in the classroom and future occupations; this, after all, is a social justice imperative. However, schools have yet to devote sufficient attention and resources to impart coding's instruction. This is generally the case in schools across the country, and Catholic schools' K–12 curriculum in the ADLA is no different. Research into the literature also did not yield any data on coding instruction in K–12 ADLA schools; therefore, my study's intent is multipronged: satisfy the dearth in literature as it relates to the ADLA, emphasize the need to integrate coding instruction, inform administrative and pedagogical decisions in K–12 Catholic schools across the ADLA to better prepare *all* students for academic and workspace demands, and consequently diversify STEM representation in both arenas.

Chapter 2 will commence with a discussion of the most crucial terms for my study, including CS, CT, and coding/programming. It will continue with an explanation of examples on how to implement its instruction in the classroom, including the challenges that will present themselves when attempting to do so. The chapter will conclude with a discussion of coding instruction's benefits, as it relates to academic achievement, occupational preparedness, and its potential to curtail the imparity in STEM representation both in school and in the workplace.

Definitions

The following section will provide definitions to and brief explanations about the terms that will be used extensively in this document. This will provide the reader additional context to what will be discussed in the proceeding sections of this document.

Computer Science

CS is "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society" (Tucker et al., 2003, p. 6). Essentially, CS encompasses various branches including coding, CT, and computing (Mason & Rich, 2019). Perry (2015) buttressed this point when referencing the Royal Society's assertion that coding is integrated in the CS discipline.

A 2015 report conducted by Google demonstrated most teachers and principals correctly identified programming and the creation of software as part of CS; however, they incorrectly deemed creating documents and presentations and researching online were elements of it as well. One factor leading to such a misconception is grade level of instruction (Google, 2015). Teachers of seventh grade and above responded more accurately to such inquiries relative to elementary instructors (Google, 2015). Furthermore, age was a contributing factor because teachers below age 50 were slightly more accurate in identifying CS concepts (Google, 2015). The study also revealed gender as a distinguishing factor, as female teachers were more likely to regard use of internet and the creation of documents applicable to CS (Google, 2015).

Computational Thinking

Many researchers and scholars have varyingly defined computational thinking (CT); thus, a single uniform definition does not exist (Rich et al., 2020). However, Papert was the scholar and mathematician who coined this term (Keane et al., 2019), and it was synonymous with being technologically well versed and able to produce algorithmic solutions to problems (Bers & Sullivan, 2019). Others since Papert's pioneering work have offered additional characterizations or elucidations of CT (Keane et al., 2019). For instance, Wing (2008) averred that CT is a type of analytic thinking connected to mathematical, engineering, and scientific thought processes. Wing (2006) expanded upon this construct by underscoring three points:

> Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. . . . Computational thinking is a fundamental skill for everyone, not just computer scientists. . . . Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. (p. 33)

Wing asserted CT enables a student to reason like a computer scientist (Mason & Rich, 2019) by exercising abstract reasoning abilities (Tonbuloglu & Tonbuloglu, 2019). Consequently, Wing called for its instruction in all schools because such abilities are of use beyond the CS domain

(Armoni, 2016). In fact, diSessa (2018) voiced Wing's (2008) possibly most fundamental declaration that CT is a universally applicable skill set and it impacts divergent disciplines. Price and Price-Mohr (2018) used Wing's (2008) interpretation and suggested computers have created a new epistemology, which is an innovative manner of thinking that can be applied in a cross-section of disciplines and curriculums.

In a similar fashion, Keane et al. (2019) defined CT as a "problem-solving methodology that involves organizing data logically and breaking down problems into parts" (p. 541). CT enables students to use and augment recursive, algorithmic, and logical thinking to articulate a resolution to a problem (Keane et al., 2019). The Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) supplemented this discussion by proclaiming CT envelops several skills including analyzing data, applying abstraction to represent the data, constructing algorithms to problem solve, and vetting the most effective and efficient means to a solution (Wong & Jiang, 2018). DePryck (2016) offered an invaluable assessment of the general nature of CT by exclaiming it centers upon the how of learning as opposed to the what.

Programming and Coding

Some scholars have used the terms coding and programming interchangeably, and others have drawn distinctions (Corradini et al., 2018). For instance, Uzunboylu et al. (2017) referred to Curzon, stating coding is essentially programming. In the broad discipline of CS, programmers write code, which are meticulous instructions for a program to function or complete a task (Hutchison et al., 2015). Coding develops skills to participate in computer programming (Duncan et al., 2014), and, instead of merely being consumers of technology, students become its producers and thus serve as active agents of their learning (Bell, 2016). In software design, the term coding is used synonymously with programming (Corradini et al., 2018; Duncan et al., 2014). In education, these words are used identically as well, and it is done so to inject interest because the word has an element of surprise or mystery attached to it (Duncan et al., 2014). Contrastingly, if one were to narrowly define coding, it only signifies entering programmatic expressions (Duncan et al., 2014). Hutchison et al. (2015) corroborated this point, stating coding originally connoted creating in a complex programming language. Such a definition simply views coding as the language that animates a program or allows a program to function. However, there are additional elements in programming, including planning and debugging (Duncan et al., 2014), and the term coding did not encompass these elements (Corradini et al., 2018).

Contrastingly, some scholars have offered a more expansive depiction of coding and are thus quite similar to characterizations of programming suggested previously. For instance, Hutchison et al. (2015) believed coding refers to the sequencing of instructions as well, and accessible and intuitive apps and programs have been formulated to allow children to learn its intricacies. Resnick and Siegel (2015) considered coding as not merely a set of skills, but also as a novel literacy, much like reading or writing, allowing for the expression of ideas, and thus, pertinent to all. Tonbuloglu and Tonbuloglu (2019) highlighted Resnick's (2012) sentiments about the power of coding exclaiming that the knowledge of coding leads to careers in computer science or software development, it develops invaluable skills such as analytic thinking, and it lends itself to collaborative work.

A code is the internal representation or realization of a program (Armoni, 2016). Without code, which are equivalent to instructions, a program simply cannot operate. There are varying understandings of coding and programming (Duncan et al., 2014); nevertheless, whether coding is defined narrowly or expansively identified as incorporating various stages such as debugging, coding websites and applications train students to become competent in these various phases during the process of coding. In fact, Resnick and Siegel (2015) exclaimed children learn more than simply coding skills with Scratch. According to Resnick and Siegel (2015), "They are learning to think creatively, reason systematically, and work collaboratively—essential skills for everyone in today's society" (p. 1). Furthermore, to buttress these points, Hutchison et al. (2015) emphasized coding applications are geared to prepare students to code computer programs by developing skills comprised of logical thinking, problem solving, and planning. Wu et al. (2020) also substantiated this notion, contending coding fosters design skills and problem-solving capacity. Specifically, coding educates students on how to simplify problems, identify errors, and generate solutions, which are capabilities inherently significant in STEM fields (Wu et al., 2020).

Statistics have borne the rising importance accorded to coding worldwide. CODE.org affirmed 30% of U.S. students and 15% of students worldwide have enrolled on their website, with numbers exceptionally escalating in recent years (Mason & Rich, 2019). Of equal importance, the number of teachers who registered on the website were a mere 10,000 toward the latter part of 2013; however, by 2018, that number had skyrocketed to 1,000,000 (Mason & Rich, 2019). A study by Soykan and Kanbul (2018) offered additional fodder to the rising importance of coding; their study indicated 70.9% of students ages 11–12 wanted to learn how to code. CODE.org became one of the principal contributors to the growth in interest among students and educators alike as they introduced the Hour of Code in 2013, allowing interested participants to engage in a coding lesson for one hour during Computer Science Education Week, which was held between December 7 and 13, 2015 (Hutchison et al., 2015). CODE.org's foundational premise is to provide all students the platform to learn programming by allowing educators to enroll students and use a variety of coding resources such as Scratch (Soykan & Kanbul, 2018).

Some scholars not only have regarded coding and programming to be synonymous, but they also have averred that coding supports CT. In fact, Uzunboylu et al. (2017) suggested one means to develop CT is through coding instruction, and DePryck (2016) offered Scratch as one resource that can be used. In fact, Estapa et al. (2018) asserted:

> These programming apps can be used to expose students not only to computer programming or coding, but they also teach mathematics concepts and the broader skills associated with computational thinking by asking students to engage in tasks that require them to do things such as group variables, apply conditional logic, develop algorithmic functions, calculate angles within geometric shapes, and more. (p. 25)

Tuomi et al. (2018) furthered such contentions by suggesting that "coding skills are seen as a combination of problem-solving, logical thinking, computational thinking, and design skills" (p. 420). Such skills, along with the ability to code, are paramount for 21st century demands according to various researchers and scholars (Tuomi et al., 2018). This, in turn, will allow students to understand the processes inherent in modern digital technologies (Tuomi et al., 2018). Jancheski (2017) referenced Tynker (https://www.tynker.com), a website teaching students to code, and corroborated Tuomi et al.'s (2018) position, positing coding is a rudimentary skill in our digital society like math and science.

Although coding inculcates a plethora of skills within students, Jancheski (2017) stated it is offered only sparingly in schools. To be sure, inroads have been made to expose a greater number of students to coding through websites such as CODE.org, which is discussed later in this paper, yet it is still generally lacking in schools. Given the assortment of skills shaped by exposure to coding, districts, dioceses, and other educational bodies should make a concerted effort by devote greater resources to it.

Implementation

STEM predominantly came into the educational fold during the Space Race in the 1950s as Congress passed the National Education Defense Act (1958), which provided funding for schools (Bers & Sullivan, 2019). However, Bers and Sullivan (2019) alluded to Bilkstein (2018) when they suggested Papert's, Feurzeig's, and their colleagues' development of Logo programming was the initial moment when CS education was advanced in educational spaces. Logo taught programming concepts to students, as it allowed them to move a turtle on the screen or even a digitized but physical one on the ground (Bers & Sullivan, 2019). Such activities or novel modalities of learning underpin the compelling point Turan and Aydoğdu (2020) raised, which is that learning the languages driving technologies are as critical as immersing oneself in traditional courses (e.g., mathematics and reading). Nevertheless, after Logo's initial appearance in educational spaces in the 1980s, its use discontinued; instead, pedagogy focused on the acquisition of CS skills, such as word processing (Popat & Starkey, 2019). Rich et al. (2017) echoed a similar position that mainstream education did not adopt computing.

Understanding and applying CT skills is best done during a child's early and formulative years (Wing, 2008). Cutumisu and Guo (2019) referenced the ISTE, stating it is crucial to integrate CT in the K–12 curriculum. Serafini (2011) echoed these overarching points by affirming it is possible to teach CT to students of all ages. Students' problem-solving skills are meager, and they are incapable of applying logical thinking in a consistent fashion (Tuomi et al., 2018); the development of CT via coding (Estapa et al., 2018; Tonbuloglu & Tonbuloglu, 2019) can improve these skills within students. In 2016, the K–12 Computer Science Framework was established through conjunctive multipronged efforts of numerous stakeholders, including the Computer Science Teacher Association, the National Math and Science Initiative, CODE.org, and various states and districts (Bers & Sullivan, 2019). This framework provides specific guidelines to be met at every level of education (Bers & Sullivan, 2019). Although this national framework establishes a path to enhancing CS, CT, and coding skills, there are states that still do not have such a framework. This fact was highlighted by Bers and Sullivan (2019), who maintained 25 states had an established set of K–12 CS standards and 10 states were in the process of constructing such a framework, but there was no evidence of the remaining 15 states having such a program or initiating steps to institute it. Furthermore, Bers and Sullivan (2019) provided a statistical byproduct of the presence and lack thereof of such frameworks by stating, on average, there are 5,000 more ScratchJr users in states that do have CS standards as opposed to states that do not. Additionally, in states that do have a CS framework, there are on average 4,605 more ScratchJr projects edited relative to their counterparts that do not have a curriculum that integrates CS (Bers & Sullivan, 2019).

Although education policies or guidelines at the national level do not necessarily need to be adhered to by states because education is a matter under the sole purview of states, CS standards incorporating coding and CT should be a stable fixture in the Common Core State Standards (CCSS; 2020). Because numerous states do not have any semblance of a CS framework and because Catholic schools and public schools in most states abide by CCSS, infusing coding instruction in this national curriculum will trigger an enhanced commitment to this shift in pedagogy in these schools. To be sure, not all states adhere to CCSS as there are 41 states currently that have adopted either the complete formulation of CCSS or a portion of it (Common Core State Standards [CCSS], 2020), with four states reneging after their initial decision. However, incorporating coding into CCSS will make significant inroads in exposing a larger number of students to coding instruction.

Bell (2016) suggested coding should not even replace or substitute any other core course in the curriculum. In fact, complementing the curriculum with CS concepts like coding serve to enhance content learned in other subject areas, such as drawing geometric angles and shapes in mathematics (Bell, 2016). Chang and Peterson (2018) voiced this sentiment as well, suggesting CT should be integrated across various traditional disciplines. Rich et al.'s (2017) study also corroborated this contention. They found educators integrated coding primarily in mathematics, but coding also was woven into other disciplines, including language and social science (Rich et al., 2017). According to diSessa (2018), instruction must shift to better attract students to STEM fields, meet the demands of modern society, engage in deeper learning, and be attune to the diversity in schools. Unfortunately, however, Rich et al. (2020) cited CODE.org in affirming most elementary schools do not impart coding instruction or CT. This affirmation is corroborated by Harlow et al. (2016), who noted although developments have taken place in infusing programming in curriculum, most elementary schools do not offer such instruction. This finding is corroborated by Project Tomorrow's (2018) research, which demonstrated 71% of school and district administrators stated coding has been infused in school, nevertheless, those experiences were limited to transient ones such as a 1-day dedication to coding during the school year or extracurricular activities. Additionally, only 7% of students from kindergarten to eighth grade reported learning coding in other subject areas (Project Tomorrow, 2018). However, coding or CT should be embedded in the school curriculum across all disciplines, so students reap its benefits (Project Tomorrow, 2018), which simultaneously circumvents the problem of finite instructional time (Harlow et al., 2016). A curriculum infusing instruction on coding software teaches students programming languages and algorithmic thinking translating into the development of problem-solving skills such as logical thinking (Kim, 2019), which are all powerful tools with which students must be equipped in a constantly altering 21st century digital society.

Pedagogy

The classroom is arguably the optimal environment to teach coding and exercise CT, as teachers can gauge student progress carefully and assist those who are in need while tasking advanced students with more challenging responsibilities. Educators devise the learning opportunities for students, and, as such, their instruction is crucial in students' acquisition of skills developed via coding (Popat & Starkey, 2019). Mason's and Rich's (2020) study demonstrated integrating coding instruction consistently and frequently improved students' attitude, confidence, and perception regarding its social value. Furthermore, CT can be developed within students at an early age across various disciplines (Bers, 2018), which potentially eases its integration in the curriculum. Wang et al.'s (2016) study suggested this same point as the overarching sentiment among teachers, principals, and superintendents in the study believed CS should be integrated in the instruction of other courses.

Coding Resources

Scratch. Scratch is the largest youth digital programming community in the world with a user base of over 21 million people composed primarily of students ages 8 to 16 (Hagge, 2018) and thus it serves as one critical means to teach students how to code. It was developed by Resnick, a pupil of Papert (Bers, 2018). It has an icon-based intuitive interface allowing for production of animations and games, among other creations (Bers, 2018). According to Calder (2018), “Scratch is a platform with symbolic, numeric, visual and aural representations that directly interact with each other. In essence, the coder uses some representations, usually symbolic and numeric, to transform or manipulate other representations, often visual or aural ones” (p. 45). Furthermore, Scratch permits students to share their creations and receive feedback (Bers, 2018). In fact, Resnick and Siegel (2015) declared the principal focus of Scratch is students’ ability to express their creativity by sharing their work with others. It is centered upon collaborative learning and is increasingly used in educational environments (Hagge, 2018). Students combine graphical blocks that control characters’ actions on the screen, making it both accessible to novice learners but sufficiently complex for advanced programmers (Grover & Pea, 2013). Scratch’s low floor and high ceiling environment challenges students at all learning levels (Grover & Pea, 2013) as it develops students’ problem-solving skills, such as logical reasoning and debugging, while simultaneously enhancing their knowledge of programming concepts, such as sequence and looping (Hagge, 2018). Scratch removes the syntactic complexity of coding (Falloon, 2016); furthermore, students can easily snap programming blocks together (Hagge, 2018) and as the created projects become more sophisticated, the underlying coding concepts become that much more intricate (Kafai & Burke, 2014). Moreover, the ease in resetting the figuration of the blocks (Calder, 2010) is yet another factor supporting accessibility. Calder (2018) continued to elucidate the workings of Scratch:

> When students write a new piece of code or change an existing piece, they might select and move blocks and input data on [the] screen directly. Moreover, if they try the code, they would see its effect on the screen, and this feedback to their inputted code would be immediate. (p. 47)

This design, free of technical complexities, makes CT and coding accessible to students and teachers (Falloon, 2016). It also facilitates the comprehension of mathematical concepts such as coordinates and variables (Jancheski, 2017). Embedded in Scratch is a do-it- yourself (DIY) culture, which promotes the formation and sharing of digital media with other community members (Hagge, 2018). Although Scratch is quite intuitive, students must possess the basic ability to read and write to use the program effectively (Bers, 2018).

The availability of ScratchEd (https://scratched.gse.harvard.edu), which is an online community of educators to share ideas and resources, collaborate, and ask questions eases the inclusion of Scratch in educators' pedagogy (Hagge, 2018). Teachers are incorporating such resources in content area instruction because there is typically no set schedule for computing classes (Israel et al., 2015). For instance, one teacher used an interactive whiteboard to demonstrate how to animate a character on Scratch (Israel et al., 2015). Students subsequently had the option to create any animation; however, they were required to use the skills exhibited by the teacher (Israel et al., 2015). Once students mastered these skills, they were afforded time for independent work with different activities of their choice without any sort of requirements or constrictions (Israel et al., 2015).

Another example illustrated how fifth-grade (Year 6) students employed Scratch to design an activity for their kindergarten (Year 1) “buddy” class (Calder, 2010). Students first conducted research on how to design the project by asking the kindergarten teacher about the content that should be incorporated into the activity and each group subsequently devised an appropriate game (Calder, 2010). Students designed a mathematics game to facilitate the understanding of numbers for their kindergarten “buddies” (Calder, 2010). This activity not only enhanced students’ intrinsic motivation by way of using Scratch, but it also developed their ability to problem solve (Calder, 2010). This example indicates the variegated opportunities that exist when using Scratch in the classroom.

Banzato and Tosato (2017) offered an additional example of using Scratch in the classroom; however, in this case, a unique twist was part of the creative process in that a robot with the Lego WeDo kit (https://education.lego.com/en-us/products/lego-education-wedo-2-0-core-set/45300#wedo-20) was to be constructed as well. Twenty students in the first level of secondary school were read Aesop’s “The Fox and the Crow,” and a discussion was held (Banzato & Tosato, 2017). Students were first asked to build the fox with the robotics kit, construct the background of the story using Scratch, and insert characters from the story in the program (Banzato & Tosato, 2017). Additionally, the robot fox was connected to a gyroscope and the gyroscope to a computer that had the Scratch program (Banzato & Tosato, 2017). The fox’s movement depended on how the groups constructed the robot, but because instructions were being transmitted from the computer with the Scratch program, students could control the fox’s movement using Scratch (Banzato & Tosato, 2017). This classroom activity proffered multiple coding opportunities for students and challenged their computational skills to fulfill the required outcomes. Scratch offers a host of creative options to produce multimedia (Jancheski, 2017), and this is one example of how Scratch can be applied in a classroom setting.

Fields et al. (2015) provided yet another application of Scratch explaining how students collaboratively illustrated a song. Each student in the group was responsible for demonstrating a portion of the song; however, it was expected that the song would be represented cohesively in Scratch (Fields et al., 2015). Characters' actions, as produced creatively by each student in the group, had to jive with other students' productions in the group (Fields et al., 2015). Fields et al. (2015) noted students accomplished this successfully with some simplistic and complex strategies. This application of Scratch serves as another example of an innovative means to teach students how to code while it is integrated into a traditional course in the curriculum.

ScratchJr. ScratchJr was developed by the combined efforts of the DevTech Research Group at Tufts University, the MIT Lifelong Kindergarten Group, and the Playful Invention Group in hopes of providing an age-appropriate and simpler program aimed primarily for students of lower age levels who are unable to read or write (Bers, 2018). Block-based languages, such as ScratchJr, represent instructions in the form of icons and eliminate the use of syntax, which can impede students' creative and cognitive development during the learning process (Bers & Sullivan, 2019). These features make coding digestible for children at that age level. Students are given the opportunity to advance their problem-solving abilities and enhance their CT skills (Bers, 2018). In fact, the foundational purpose in designing this program was not to necessarily engender the development of future programmers and software developers but instill crucial skills within students to equip them with the tools to partake in their communities and society (Bers, 2018). As of February 2018, with an average user base numbering 104,000, students using ScratchJr had produced 20 million projects (Bers, 2018). Bers (2018) highlighted the functionalities involved in the use of ScratchJr:

> ScratchJr has a palette of programming blocks, a user's library of projects, a main project editor, and tools for selecting and drawing characters and background graphics. Children drag blocks from the palette into the scripting area and then snap them together to create programs that are read and played from left to right. The programming blocks are organized into six categories represented by different colors: yellow Trigger blocks, blue Motion blocks, purple Looks blocks, green Sound blocks, orange Control flow blocks, and red End blocks. When put together as a jigsaw puzzle, these programming blocks allow children to control their character's actions on the screen. (p. 3)

Furthermore, students can add text and speech and use the paint editor option as an additional form of flexibility to modify their characters in the story and generate imaginative, creative, and complete stories (Bers, 2018). Students have a wide latitude of imaginative and artistic options, including the possibility to create animal collages and games (Bers & Sullivan, 2019). As students are engaged in this process, they are being exposed to vital CS skills (Bers, 2018), such as algorithms and debugging (Bers & Sullivan, 2019). In fact, "ScratchJr users are encouraged to learn by experimenting and by making mistakes, by fixing their bugs and by problem solving" (Bers & Sullivan, 2019, p. 120).

Falloon (2016) demonstrated a concrete example of the use of ScratchJr in the classroom. Students in kindergarten and first grade were tasked with creating geometric shapes such as squares and triangles along with certain upper- and lower-case letters (Falloon, 2016). Students were seen to not only exercise skills such as predictive thinking as they collaboratively, in pairs, hypothesized the result of their coding, but also evaluative thinking as they juxtaposed the outcomes with the required criteria (Falloon, 2016).

Logo. Bers and Sullivan (2019) referenced Bilkstein (2018) in crediting Papert, Feurzeig, and their colleagues for creating the programming language known as Logo (Logo Foundation, n.d.). Logo's language is based upon reduced instructions that manifests itself in creating syntax with ease (Serafini, 2011). It does not depend on a drag-and-drop approach and instead permits the student to type the syntax (Serafini, 2011). Possibly even more significant than the specific skills Logo develops within students is Jancheski's (2017) statement that it champions student autonomy and individuality.

Programming Games. Programming games expose students to sequencing, problem-solving, and logical thinking (Bers & Sullivan, 2019). The games are already developed, and students exercise the aforementioned skills to complete various tasks (Bers & Sullivan, 2019). For instance, Lightbot prompts students to provide directional instructions to a robot to activate the blue tiles and light the bulb (Bers & Sullivan, 2019). Kodable is a maze-based game, and Cargo Bot prompts students to move crates to accomplish the given task (Bers & Sullivan, 2019). These games exercise students' logical and problem-solving skills and serve as alternate forms of learning.

TouchDevelop. Microsoft's TouchDevelop (https://www.microsoft.com/enus/research/project/touchdevelop/) is a free online interactive program that allows users to not only create and run programs but develop applications on devices such as smartphones (King, 2015). The program is intuitive in that there are three skill levels, beginner, coder, and expert, which allows students to seamlessly progress from drag-and-drop blocks to text- or syntax-based programming while being guided by tutorials (King, 2015). Students can progress at their own rate of comfort, but it provides learning opportunities for novices and the well-versed alike. King (2015) is a teacher who used TouchDevelop in the classroom. Students were initially asked to establish an account and then tasked with completing a beginner project called *First Steps With Turtle* (King, 2015), which permitted students to create a rudimentary application with some colorful and geometric drawings. The tutorial lucidly explained how the code resulted into specific actions by the turtle (King, 2015). As students progressed in their learning, they were tasked with calculating angles of movement for the turtle (King, 2015). Students engaged in a trial-and-error process to devise solutions to the problems they encountered; ultimately, they reported loving the activity and appreciating the flexible nature of the project as it allowed them to work at their own pace to bring about desired outcomes (King, 2015). Moreover, another crucial factor in their evaluation was the ability to discuss and resolve issues relating to their coding collaboratively (King, 2015).

Tynker. Hutchison et al. (2015) wrote, "Coding apps . . . support students' understanding of if–then relationships, learning how to navigate informational text, and experiencing the importance of sequencing events correctly" (p. 498). The if-then commands serve as instructions for creations such as animations (Hutchison et al., 2015). Tynker (https://www.tynker.com) is a coding app that educates students on the use of if-then commands; although its interface is like Scratch's and ScratchJr's, the program is inherently more intricate (Hutchison et al., 2015). Additionally, Tynker offers preprogrammed games to be used to learn coding skills in addition to more creative avenues such as story construction (Hutchison et al., 2015). Students can either incorporate available templates with scenes and characters or construct these elements themselves (Hutchison et al., 2015).

Robotics. Robotics and coding applications are relevant and budding sources of technology in education (Cakir & Guven, 2019). In fact, "significant overlaps among coding, CT, and robotics suggest that robots are frequently being used to teach coding or programming, and coding, programming, and robotics can effectively be used to teach CT" (Mason & Rich, 2019, p. 799). Thus, Keane et al. (2019) cited (Bers et al., 2014) in explaining that robotics has become increasingly used for STEM subjects and programming in schools. The emphasis to instruct coding is part and parcel to developing 21st century skills and thus, such capacities are being emphasized in STEM disciplines including robotics (McCoy-Parker et al., 2017). Additionally, "collaborative projects like the programming of a robotic performance gives students experience working in teams, problem-solving, adaptability, managing time constraints as well as the opportunity to present their work to others, all skills that will be valued by future employers" (McCoy-Parker et al., 2017, p. 102). Williams et al. (2020) noted coding's impact on perseverance by explaining the general frustration of fifth–grade students when they initially participated in a coding club; however, the students exhibited perseverance and continued interest (Williams et al., 2017).

Papert steadfastly maintained robotics can augment student learning (Keane et al., 2019). Robotics motivated students to explore CT in an engaging and interactive manner (Grover & Pea, 2013). Bers and Sullivan (2019) exclaimed robotics fosters cognitive and fine motor skills as students interact with sensors and motors to create a product or project. Keane et al. (2019) alluded to Mubin et al. (2013) in affirming robotics even possesses the capacity to teach new languages. In fact, Keane et al.'s (2019) research demonstrated how students used robotics to learn the native language known as Narungga in Australia:

> During this study, students were asked to program the humanoid robot to move around a maze with Narungga place names. Another group of students programmed the robot to recognize drawings of local animals and respond with the correct name in Narungga. Further, the students programmed the robot to recognize paintings of animals and then orally express them in the Narungga language. (pp. 535–536)

Once the students executed the coding, those instructions were transferred to the robot allowing it to stand, sit, speak, and perform other functions (Keane et al., 2019). The presence of the robot in the classroom infused further interest in the activity of coding (Keane et al., 2019).

Robotics uses block-based programming permitting students to simply drag and drop instructions, which translate to specific functions for the robot (Cakir & Guven, 2019). Consequently, students can visualize the tangible effects of their coding sequences on the robots (Cengel et al., 2018) and modify the instructions immediately to bring about desired outcomes. Students, in this trial-and-error process, are excited to see their progress and motivated to reach the end goal. The research conducted on the effect of robotics and coding on the acquisition of the Narungga language demonstrate these points. Schools can establish a wide-reaching robotics program, using age-appropriate kits, to allow students from several grade levels the opportunity to experience robotics programming.

Many of these kits are possible to program with a variety of devices, such as iPads or Windows based devices. The "Blue-Bot," for instance, is Bluetooth compatible, functional on tablets and computers, and students can specify algorithms on a tablet that are subsequently communicated to the robot (Bers & Sullivan, 2019), allowing it to perform the designated functions. Another robotics kit is the Lego Mindstorms EV3 (https://www.lego.com/en-us/product/lego-mindstorms-ev3-31313), which also uses the drag-and-drop platform (García-Peñalvo et al., 2016) to teach students coding and programming. “The visual programming environment supports basic programming structures such as loops and conditional statements” (García-Peñalvo et al., 2016, p. 23) and can be used in the elementary and even the university school level.

Robotics kits are created with different programming languages and varying functionalities. According to Cakir and Guven (2019), the microcontroller Arduino-based robotics and coding resources are a straightforward means to learn how to program, as it is functional with the drag-and-drop system and uses various sensors. Arduino (https://www.arduino.cc) is “an open source hardware and software for building digital devices and interactive objects that can sense and control objects in [the] physical and digital world” (Rossano et al., 2018, p. 264). Arduino is principally controlled with memory-based programmable microcontrollers and microprocessors (Rossano et al., 2018). Once the microcontroller’s memory is loaded with instructions, it is directed to the microprocessor to effectuate the commands (Rossano et al., 2018). An additional stunning element of Arduino is its sensors, which can measure light, temperature, carbon monoxide, and even radioactivity, among other phenomena (Rossano et al., 2018). Such capabilities immerse and motivate students in the learning process (Rossano et al., 2018) and are another means to teach coding.

One example of using Arduino in the classroom centered upon sixth-grade science instruction on the pulse of a heart (Cakir & Guven, 2019). Students used a resource called mBlock program that converts block codes into C++; therefore, students need not write syntax-based codes (Cakir & Gulven, 2019). The mBlock program was connected to the Arduino microprocessor and the block-based instructions were transferred to the processor (Cakir & Gulven, 2019). Thereafter, a pulse sensor was connected to the microprocessor; students placed their hands on the sensor, monitored the flashing red lamp to experience the results of their coding, and calculated measurements of the pulse (Cakir & Gulven, 2019).

Maker. Maker activities is a method of designing and crafting objects by combining different techniques, such as robotics, which serves as the digital component in the overall work and woodworking, operating as one traditional mode of crafting (Tuomi et al., 2018). Calabrese-Barton and Tan (2018) cited the claims of (Kafia et al., 2014) that maker activities in the form of e-textiles combine digital functions (e.g., coding of microcontrollers) and creation of circuits with conventional forms of constructing (e.g., sewing). Such activities also focus on participatory learning as students share ideas and methods used to construct their objects electronically and physically (Tuomi et al., 2018). Arguably of primary importance, maker activities can be a source of motivation and engagement on part of students in STEM subjects (Tuomi et al., 2018). Indeed, maker activities have gained momentum not only in the United States but also in other countries in the world (Calabrese-Barton & Tan, 2018). For instance, Finland has invested time and resources in employing maker activities in the classroom, especially because Finnish schools have always embraced a hands-on approach to building (Tuomi et al., 2018). Furthermore, Finland's state policy regarding education is to develop an information society (Tuomi et al., 2018). These educational objectives are supported by the maker activities.

A specific maker activity is electronic textiles (e-textiles), and codeable circuits, which are part and parcel to e-textiles, integrate various skills including, but not limited to, engineering and computing (Litts et al., 2017). This form of maker activity was used in a high school junior elective class of 23 students, mostly comprised of female students (Litts et al., 2017). The students took 15 sessions, each lasting 90 minutes during a 3.5–week period and were responsible for tasks such as coding microcontrollers so they could properly turn on and off LED lights in patterns (Litts et al., 2017). At the end of this course, students' knowledge of coding expanded significantly as students could read, design, and remix code as applied to circuitry (Litts et al., 2017).

Lui et al. (2020) conducted a study depicting the creation of e-textiles as well. Lui et al. focused on the impact of paired programming in the process of constructing an animated sign with the school's name and they specifically assessed how students designed, crafted, and coded their work. Each pair was tasked with creating a specific letter in the sign (Lui et al., 2020). Paired programming is based on a collaborative relationship between students whereby one serves as the navigator, who provides instructions, while the other student is the driver who executes the instructions (Lui et al., 2020). Using this system of learning programming has been successful in K–12 schools (Lui et al., 2020). In fact, Litts et al. (2017) found groups collaborated in deciding the aesthetic and electrical components of the work; however, groups working cohesively throughout the process were naturally effective at problem solving and generally more productive.

Coding Opportunities Beyond School

Students can expand their knowledge of coding and concomitantly exercise CT skills by availing of online resources at home or even take part in clubs. Of course, teachers do not have as much ability to manage student progress, unless they are combined with specific class requirements, but students who are especially interested, have an assortment of options to practice coding. For instance, Microsoft Virtual Academy or Codecademy (https://www.codecademy.com) are virtual forums that enhance curiosity and advance coding skills (Tuomi et al., 2018). Furthermore, the CoderDojo (https://coderdojo.com) is a free online resource, primarily led by volunteers to teach children between the ages of 7 and 17, how to build websites and create apps and games (Hagge, 2018). Another support, geared toward female students, is called Girls Who Code, which hosts summer camps and after-school clubs to teach intricacies of the coding process (Hagge, 2018).

International Efforts

Countries have adopted varying policies on how to impart coding instruction. Many countries in Europe restructured their curricula to impart coding instruction (Tuomi et al., 2018) For instance, schools in Estonia offered programming education to students from the age of seven starting in fall of 2012 (Uzunboylu et al., 2017) and such efforts are primarily geared to bolster the waning supply of capable programmers (Perry, 2015). The United Kingdom instructs basic programming skills and processes to store and organize data to students from the age of 5 (Uzunboylu et al., 2017). Bers (2018) bolstered these points, citing Livingstone (2012), who averred those 16 countries, including Spain, Malta, and France, had incorporated its instruction in the curriculum. Israel has an expansive CS program with it being its own solitary course from 1995 at the post-primary level (Perry, 2015).

In Finland, primary teachers are expected to impart coding in grades first through sixth by connecting it with other courses such as mathematics (Tuomi et al., 2018). The mandate to do so originated in 2016 and this is partly attributed to Finland's high esteem for coding as it is on par with reading and writing (Wu et al., 2020). To prepare teachers for this significant shift, a community-based online course was created, and 500 teachers volunteered to participate in it (Rich et al., 2017). Teachers from lower elementary levels became proficient at using ScratchJr and older elementary grade level educators were trained using Scratch (Rich et al., 2017). Over 70% of teachers concurred with the mandate, suggesting programming was necessary to instruct in primary schools (Rich et al., 2017). Due to the program, teachers' general apprehension centered upon teaching coding diminished and their assessment on coding's accessibility improved (Rich et al., 2017).

Bers (2018) referred to the Australian Curriculum and Assessment Authority by positing non-European countries such as Australia, Malaysia, and Singapore added development of CT skills in their K–12 curricula. In most Australian provinces, digital technology is compulsory in K–10 public and Catholic schools, an imperative rooted in the perceived importance of developing CT and coding skills (Williams et al., 2020). Wu et al. (2020) corroborated these affirmations, stating Singapore, China, and Finland have refocused their curriculums to include higher order thinking and instilling of 21st century skills. These three countries also happen to be among the top performers on Program for International Student Assessment (PISA) indicators (Wu et al., 2020). China took steps to revise its national curriculum by emphasizing CT skills at the high school level (Wu et al., 2020). Furthermore, schools in Singapore begin coding instruction with preschool children (Wu et al., 2020). Seow et al. (2017) referred to the Smart Nation Programme in Singapore, which was instituted in 2014, to technologically prepare citizens to participate in a variety of sectors, including government and business. Coding and CT skills are focused upon in this national initiative even for preschool children (Seow et al., 2017). Engendering interest at this early age is a key strategy in the program; students in primary and secondary schools are introduced to Scratch and various robotics kits such as Lego WeDo (Seow et al., 2017).

Rich et al. (2017) underscored efforts undertaken by Scotland to prepare teachers to instruct computing. PLAN C, which was subsidized by the government, compensated teachers during their 6- to 9-week preparation program in addition for the meetings they held to discuss incorporating computing strategies in the classroom (Rich et al., 2017). Ninety percent of teachers responded positively about the meetings and reported increased confidence in teaching computing (Rich et al., 2017).

U.S. Efforts

The focus on programming has been resuscitated in recent years, as is evinced by the K–12 CS guidelines (Popat & Starkey, 2019); thus, states have taken varying steps to implement a curriculum based on these principles. In fact, Mason and Rich (2020) referenced CODE.org in highlighting the number of states that adopted some formulation of K–12 CS framework amplified from 14 to 44 in a span of 5 years dating from 2013 to 2018. States have increasingly received guidance from diverse institutions, including the International Society for Technology in Education (ISTE). Mason and Rich (2019) referred to the ISTE and its efforts to reconfigure

CT standards to be applied across various disciplines rather than solely in a CS course. Specific guidance on how educators can cultivate CT is underscored (Mason & Rich, 2019), which is a key component in delivering such content in the classroom. Mason and Rich (2019) also cited the K–12 CS framework and its five overarching concepts (e.g., algorithms, programming, data and analysis) to assist states and local governments in propagating CT in schools.

The Code.org Advocacy Coalition (2019) highlighted nine principles, such as the need to create programs in institutions that prepare preservice teachers and securing funds for teacher preparation and development to assist states in implementing CS. According to the Code.org Advocacy Coalition (2019), "Although computer science is broader than programming, some direct programming experience is integral to learning the fundamental concepts and is used as a defining characteristic to differentiate foundational computer science courses from non-computer science courses" (p. 88). In 2013, only 14 states had implemented at least one of these policies and the number expanded to 50 since (Code.org Advocacy Coalition, 2019). Studies revealed states incorporating more of the nine policies tend to have a greater percentage of high schools that instruct CS (Code.org Advocacy Coalition, 2019). In fact, Arkansas, Idaho, Nevada, Indiana, and Maryland have adopted all nine policies (Code.org Advocacy Coalition, 2019), which eased the integration of CS courses in the classrooms across each state. One impetus for these nine policies was to generate equitable access to technology in the classrooms. The Code.org Advocacy Coalition (2019) wrote:

> At the heart of the national movement is the goal of increasing the participation of historically underrepresented groups. The principle of equity and diversity is an integral part of each of these nine policies; for example, some states prioritize funding for districts

> who make focused efforts to engage underrepresented groups, and other states call for annual report of student demographics in computer science courses that schools are required to offer. (p. 5)

As additional evidence of states' efforts, 34 states have developed CS standards based on the K–12 Computer Science Framework and the Computer Science Teachers Association K–12 Standards, with 5 in the process of completing this task as well (Code.org Advocacy Coalition, 2019). Nevertheless, as of 2019, only 13 states, including California, Georgia, Maryland, Utah, and Indiana, among others, have taken tangible steps to implement it. States are increasingly aware of the connection between CS and equity and, consequently, have taken such steps to mitigate the gap. However, much work remains across the country to provide these opportunities equitably to all students.

Perceptions of Challenges to Implementation

Numerous obstacles either prevent or dissuade schools from integrating coding or CT instruction in schools. First, a dearth of resources, such as devices (Mason & Rich, 2019; Rich et al., 2017) and a lack of updated technology (Yadav et al., 2016) or internet connectivity impedes coding's instruction (Rich et al., 2017; Rich et al., 2020). Second, educators are not trained properly to integrate its instruction in the curriculum (Rich et al., 2020). Third, they lack confidence in the subject matter (Sentance & Csizmadia, 2016) and possess a low perception of self-efficacy, even with in-service sessions (Rich et al., 2020). A study conducted by Kong and Wong (2017) highlighted their lack of content knowledge and illustrated the trepidation of some educators due to their perceived limited level of cognition as compared to their pupils. Fourth, even if educators possess the essential skills to teach coding, some teachers may not necessarily believe it is warranted to do so given the finite time they hold to instruct traditional courses (Duncan et al., 2014; Rich et al., 2017, 2020). Sentance and Csizmadia (2016) explained congestion in the curriculum is an extrinsic challenge that frustrates teachers. Finally, Yadav et al. (2016) explained how the inefficient and ineffective teacher certification process and lack of opportunities hinder computing's instruction in schools.

Mason and Rich (2019) drew attention to the reality that educators are pedagogically and emotionally unprepared to instruct computing due to lack of training. As case in point, survey data from the National Assessment of Educational Progress indicated less than 30% of students stated their teachers facilitated the development of higher order thinking skills (Wenglinsky, 2015). This was due to the reality that most elementary and middle school teachers are not trained in computer use (Wenglinsky, 2015). Sentance and Csizmadia (2016) also suggested teachers must be pedagogically trained to instruct CS. Kong and Wong (2017) buttressed these points, declaring educators do not possess the content or pedagogical foundations to instruct CS. CS courses and CODE.org activities have increased in classrooms; however, schools continue to experience difficulty in hiring qualified CS teachers (Wang et al., 2016; Yadav et al., 2016). Additionally, Williams et al. (2020) stated most countries generally have difficulty finding capable teachers who can instruct coding. Rich et al.'s (2017) study revealed more than 55% of teachers were unprepared to instruct coding before their decision to do so. A staggering 75% of teachers from the United States who were part of the study indicated they had no prior experience with coding or simply had rudimentary exposure to it (Rich et al., 2017). Consequently, Yadav et al. (2016) suggested teachers not only need to be taught computing concepts but also their delivery of such concepts need to be developed. Barr and Stephenson

(2011) and Bell (2016) similarly exclaimed teachers not only need resources to enhance their knowledge regarding CT, but to successfully shift to an altered pedagogy that incorporates it as well. For educators who are already teaching, professional development sessions are warranted to effect actual educational change (Barr & Stephenson, 2011). However, to supplement these efforts, resources such as curricular materials and the presence of professional learning communities headed by educators who have CT experience are paramount (Barr & Stephenson, 2011). Sentance and Csizmidia (2016) supplemented this point by suggesting teachers need technical solutions by the school's information technology department. Preservice programs must also feature instruction on CT across various disciplines as well (Barr & Stephenson, 2011) to better equip teachers to integrate it in the curriculum.

To ameliorate this situation, Mason and Rich (2019) echoed various scholars' and researchers' sentiments by underscoring the need to expose preservice and seasoned teachers to professional development sessions so they can instruct programming competently. Indeed, Seow et al.'s (2017) study indicated teachers, whose backgrounds were not in CS, were willing to and did participate in such sessions and even collaborated thereafter to continue harnessing their pedagogy in this regard. Furthermore, Williams et al.'s (2020) study demonstrated, among the 10 teachers who participated in a coding preparation program, eight responded to a survey positing they were prepared and confident to instruct it.

Wang et al.'s (2016) study demonstrated the firm belief among teachers, principals, and superintendents that CS instruction requires extensive training. Moreover, a crucial element of comprehensive preparation goes beyond the practicalities involved in augmenting teacher knowledge and pedagogy; molding teachers' belief system regarding the imperative to instruct coding and enhancing their sense of self-efficacy hold primacy as well (Rich et al., 2020). After all, a teacher's deleterious disposition regarding coding will feasibly impress upon students (Rich et al., 2020). According to Rich et al. (2020), "Self-efficacy beliefs refer to an individual's confidence to perform a role or task within a specific domain" (p. 4). Therefore, self-efficacy is not necessarily an accurate gauge of teacher effectiveness; rather, this notion centers on their perception (Rich et al., 2020), which may be faulty to a degree. Indeed, it is distinctly possible that a teacher may have low CT self-efficacy, yet instruct it quite capably (Rich et al., 2020). Rich et al.'s (2020) study showed that teachers were able to learn computation thinking skills and implement them in the classroom. This example demonstrated that with proper resources and time, teachers can implement lessons and activities that they did not have any experience with in the past.

Wu et al.'s (2020) study highlighted efforts related to coding and CT instruction in five countries and buttressed the chief points Mason and Rich (2019) raised: Most educators do not have the capacity to instruct coding and thus, and they contend education and professional development sessions on pedagogy should be offered to teachers to counteract this deficiency. Rich et al.'s (2017) research validated this notion as it indicated educators felt diffident about their coding knowledge and their capacity to instruct it. Perry (2015) exclaimed identical assertions by stating teachers do not receive adequate preservice or continuing education to effectively teach coding. In fact, among 231 teachers registered for the General Teaching Council in Northern Ireland, only 0.9% specialized in instructing computing or CS. However, Mason and Rich (2019) offered an optimistic assessment on the impact of professional development with the following: "The studies we reviewed indicate that training and PD can help elementary school teachers to overcome their knowledge, attitude, and efficacy barriers" (p. 807). Moreover, ephemeral preparation is insufficient; rather, such efforts must be enduring to capably prepare educators to teach coding, CT, or robotics (Mason & Rich, 2019).

Chang and Peterson (2018) conducted a study demonstrating the importance of teacher preparation for the instruction of CT as well and accentuated how contact with coding resources possesses the powerful potential to shift educators' minds regarding its significance. Upon becoming more familiar with CT through coursework, 44 preservice teachers were able to connect CT with their own beliefs and experiences on education and their perception of coding being solely related to CS evolved to comprehending CT's importance in developing various skills (Chang & Peterson, 2018). After their experience with coding, originally skeptical teachers admitted to coding's importance to the development of life-long skills within elementary level students (Chang & Peterson, 2018). Teachers were additionally given a reflection prompt as part of the study, and one participant explained she accessed coding resources to teach CT (Chang & Peterson, 2018). Participants additionally displayed interest in coding resources such as Black Girls Code and Girls Who Code, and one teacher was awed by the number of girls who were coding and how it holds the potential to empower women (Chang & Peterson, 2018). Other participants as well commented on the social justice ramifications of coding and CT because their instruction would enlarge pathways for future successful careers for historically marginalized populations (Chang & Peterson, 2018). These findings are generally corroborated by Cutumisu's and Guo's (2019) study, which indicated preservice teachers did not believe they had the requisite knowledge or adeptness to instruct CT; however, they demonstrated alacrity to incorporate it in their pedagogy. As such, Cutumisu and Guo recommend university teacher preparation programs use coursework available in CODE.org and additionally prepare preservice teachers to embed CT skills in their pedagogy. Mason and Rich (2019) and Seow et al. (2017) also echoed various scholars' and researchers' sentiments by underscoring the need to expose preservice and seasoned teachers to professional development sessions so they may competently instruct programming. Indeed, Seow et al.'s study indicated teachers, whose backgrounds are not in CS, were willing to and did participate in such sessions and even collaborated thereafter to continue harnessing their pedagogy in this regard. Furthermore, Williams et al.'s (2020) study demonstrated, among the 10 teachers who participated in a coding preparation program, eight responded to a survey positing that they were prepared and confident to instruct it.

Yadav et al. (2016) casted ire on the teacher certification process as an additional impediment to teaching computing concepts in the classroom. An inadequate number of teacher certification programs offer CS certification and those that do not prepare teachers to instruct it in the classroom (Yadav et al., 2016). CS is generally confused with technology education or instructional technology (Yadav et al., 2016); therefore, potential educators are not trained properly. Nevertheless, although these teachers lack CS content and pedagogical knowledge, they are expected to instruct CS courses and are generally tasked with instructing non-computing courses as well, leading to a high attrition rate (Yadav et al., 2016). Research suggested these novice teachers felt isolated and experienced discomfort due to teaching beyond their area of expertise and inadequate planning time (Yadav et al., 2016).

Yadav et al.'s (2016) study focused on 23 teachers' perceptions and highlighted their various challenges in instructing CS in the classroom. For instance, as aforementioned, due to the teachers' inadequate content knowledge, they were obligated to learn the subject matter while simultaneously attempting to assist students; however, these dual and conflicting tasks rendered them incapable of meeting students' programming needs CS courses (Yadav et al., 2016). Some teachers in the study attributed this difficulty to inadequate teacher preparation programs (Yadav et al., 2016). Teachers also identified pedagogical challenges; due to the student-centered nature of CS class, teachers experienced difficulty in maintaining consistent student engagement (Sentance & Csizmadia, 2016; Yadav et al., 2016). Sentance and Csizmadia's (2016) study also yielded a lack of student patience and resilience as challenges. Furthermore, because students need extended assistance, the teacher-to-student ratio made it difficult to properly tend to all (Yadav et al., 2016). Additional pedagogical challenges observed by teachers included proper evaluation; because CS assessment tools are scarce, teachers did not believe they accurately gauged student achievement (Yadav et al., 2016).

In addition to content and pedagogical shortcomings, teachers in Yadav et al.'s (2016) study emphasized the lack of community with CS instruction. Because CS teachers are very limited on a campus or across schools in a district or a specific area, surveyed teachers expressed a sense of isolation (Yadav et al., 2016)

Benefits

President Obama's justification for his 2016 initiative centered upon developing CS and CT skills within students (Price & Price-Mohr, 2018). He did not want students to simply be consumers of technology; rather, he wanted them to better understand its underlying principles, so they could become future producers of new technology (Price & Price-Mohr, 2018). Bell (2016) echoed President Obama's sentiment stating one impetus to impart coding instruction was to develop students' ability to create software as opposed to merely consume it. Consequently, students would be free of the shackles of dependence on what others produce (Bell, 2016). Resnick and Segal (2015) provided Scratch as an example to accomplish this goal, asserting, "Scratch community members begin to see themselves as creators who are able to make their own projects on the computer, not just interact with programs created by others" (p. 6). Resnick and Segal (2015) even likened coding to how esteemed Brazilian philosopher Freire regarded writing as a form of literacy, especially crucial for underserved communities. Depryck (2016) stated:

> Introducing coding in the curriculum is really not about coding itself. It is, rather, about introducing a culture of algorithmic thinking, breaking down more complex actions into a sequence of instructions, and computational thinking, focusing on problems and their solutions. This computational thinking is itself based on a set of metacognitive strategies in a wide array of domains, applicable beyond coding. (pp. 28–29)

Indeed, coding develops CT skills (Estapa et al., 2018), which are in consonance with President Obama's initiative, as those who are taught to code will be able to produce new software, programs, and applications in the future. In fact, "It is the iterative experience of software coding, that increases algorithmic thinking, problem solving, and logical thinking; it can provide various opportunities for developing students' computational thinking ability" (Kim, 2019, p. 70). Hutchison et al. (2015) resounded Gee's (2013) powerful words that coding applications result in deep and consequential learning for students; in fact, "coding apps can also help students practice logic, reasoning and problem solving to create or manipulate digital content through the use of computational thinking" (p. 500). Rich et al.'s (2017) study noted the development of problem-solving skills as one of the chief reasons why educators deemed integrating coding in the curriculum as crucial. Additionally, students exert a concerted effort to resolve their errors, which demonstrates logical thinking (Calder, 2018), which simultaneously results in a sense of accomplishment (Kim, 2019). Rich et al.'s research also demonstrated a high level of student satisfaction and fulfillment in addition to enhanced confidence in academics due to adeptness in coding. Also, students who struggle in traditional courses (e.g., language and math) found their niche with coding instead (Rich et al., 2017).

Coding instruction has a direct impact on students' future vocations as some will pursue disciplines related to it. Although some students will not necessarily strive to attain careers related to coding, acquiring such knowledge is significant regardless given society's reliance on technology. Data suggested infusing coding in primary school curriculums inculcates a plethora of skills, including computational and higher order thinking skills (Falloon, 2016). Additionally, Falloon (2016) asserted coding activities engenders creative self-expression, collaborative engagement, and an ability to critically assess technologies. In addition to producing tangible benefits in the classroom and future workplaces, offering such instruction will also make headways toward producing a more socially just society. If *all* students are offered such instruction in schools, regardless of gender, background, or any other personally distinguishable characteristic, students will be better equipped to meet the challenges of the classroom and employees from wide segments of the general population will have a more equitable opportunity to pursue technology related careers.

Academic Achievement

Falloon (2016) referenced Wing in stating CT is a 21st century literacy providing educational benefits, such as the capacity to analyze and problem solve, which are transferable to any field of study or discipline. Thus, CT opportunities must be furnished to students at an early age (Estapa et al., 2018). Serafini's (2011) study demonstrated how students were forced to take their breaks because they did not want to stop programming. Ultimately, CT will be used by students regardless of their professional careers (Duncan et al., 2014), and, because CT can be developed via coding (Estapa et al., 2018), it is pragmatic to combine educational objectives with technology to have the most fruitful of student outcomes. Coding facilitates active learning, described as students' direct role over the learning process (Popat & Starkey, 2019), which arguably leads to greater student achievement.

In adjunct to cultivating general CT skills, coding instruction eases the grasp of mathematical concepts, including calculating angles and grouping variables (Estapa et al., 2018). Mason and Rich (2020) affirmed studies have demonstrated improved achievement in math and problem-solving abilities because of coding instruction. Mason and Rich (2020 cited Rich et al. (2013) in stating that programming instruction improved performance in mathematics and enhanced problem solving capacities. Calder (2018) highlighted students' development of mathematical concepts such as angles and measurements of time and length in his research. Duncan et al. (2014) validated this claim as well, asserting content such as coordinate systems and negative numbers are learned due to programming via Scratch. Calder (2018) stated, "Students could actively experiment with angle size, for example, in ways that would not be possible without the digital medium" (p. 55). Popat and Starkey (2019) corroborated these sentiments as well by citing various studies demonstrating the advancement of problem-solving skills as applied through mathematical content during the process of coding. Additionally, students' confidence in math and science concepts improved by participating in robotics activities (McCoy-Parker et al., 2017). As an additional example of coding's connection to mathematics, students who struggled during conventional lessons were able to competently explain math concepts while coding as part of a robotics activity (McCoy-Parker et al., 2017). This example is illustrative of coding's potential to enhance student achievement in traditional courses.

Studies have also shown coding's impact on achievement in other subject areas. For instance, coding allowed students to better understand concepts in art history (Popat & Starkey, 2019). Cakir and Guven (2019) asserted Arduino-assisted coding applications assisted in comprehending science content. Specifically, students better comprehended the concept of pulse in the circulatory system by seeing the contraction and relaxation of a device powered by Arduino (Cakir & Gulven, 2019). This robotics and coding activity connected real-life concepts and science (Cakir & Gulven, 2019). Furthermore, Hutchison et al. (2015) believed coding develops early childhood literacy by addressing various English language arts standards, such as producing coherent writing, collaborating with others, expressing ideas, and learning general academic and technical terminology that will contribute to effectively reading, speaking, and writing at the college and career levels. Barr and Stephenson (2011) provided additional support to the contention that CT skills such as data collection and data analysis can be enhanced in various disciplines including social studies, science, and language arts. Tran (2019) also supported this contention, stating there is evidence to integrate CT into disciplines such as history, language arts, and science. For instance, regarding data collection in social studies, students can study battle statistics or population data (Barr & Stephenson, 2011). In science, students can collect data from an experiment as they engage in linguistic analysis of sentences in language arts classes (Barr & Stephenson, 2011). In reference to data analysis, students can identify patters in the data that they collected in social studies class and analyze the data from a science experiment (Barr & Stephenson, 2011). Furthermore, Wenglinsky's (2015) study demonstrated students performed better on the NAEP U.S. history assessment when students used computers to complete projects, communicate, and create charts, tables, and graphs. These tasks inculcated critical thought and problem solving, which demonstrate the potential for technology when used effectively.

An important element to also consider in the analysis of coding instruction on improved results is student self-efficacy. Banzato and Tosato (2017) gleaned from their studies, based on students constructing a robot fox and using Scratch to operate it, that partaking in coding increased students' sense of self-efficacy by a small degree. Soykan's and Kanbul's (2018) study demonstrating that students who took a coding course had a higher sense of self-efficacy relative to students who did not. Mason and Rich (2020) provided additional endorsement to this overarching claim by stating computing courses enhanced students' attitudes toward STEM subjects.

Keane et al. (2019) highlighted student curiosity as a different byproduct of coding. Their study focused on the programming of a humanoid robot that had sensors for hearing, sense, and touch. Students were engaged and motivated to learn more about programming to code various functionalities for the robot (Keane et al., 2019). However, students were first tasked with learning Narungga, an indigenous Australian language, and subsequently programming the robot in that language (Keane et al., 2019). Various CT skills, including debugging and algorithmic reasoning, were developed (Keane et al., 2019). This experiment not only effectively taught students a new language, which prior traditional attempts had failed to do (Keane et al., 2019), but it also simultaneously enhanced their ability to program. This is yet another innovative example of how to embed programming in curriculum.

Falloon's (2014) study of Year 1 and Year 2 students using ScratchJr revealed inserting coding into the curricula provides teachers with a tool to develop computational and higher order thinking skills within students. Turan and Aydoğdu (2020) stated, "Robotic and coding education in all students from preschool to higher education . . . can gain great importance in roles such as increasing cognitive achievements, improving the learning process, gaining and supporting academic skills" (p. 4361). Wenglinsky (2005) stated use of technology by teachers to advance higher order thinking skills is especially in demand by employers today because it can lead to complex problem solving through creative means. Collaborative and self-management skills are additionally ripened because of using Scratch (Falloon, 2016). Calder (2010) affirmed this assertion by stating collaboration and logical reasoning are advanced due to Scratch. In fact, Fields et al. (2015) affirmed, because of collaborative work, each student took ownership over the entire product and realized the work could not have been accomplished individually. Popat and Starkey (2019) provided additional credence to this contention by stating coding develops collaborative skills, indicating students can effectually communicate and work with others. Lui et al. (2020) conducted research on paired programming, a form of collaborative work, and stated, "The high level of social engagement generally seems to fit naturally with more shared work approaches. The work of sharing tasks and engaging with one another's individual work necessarily involves active discussion and communication" (p. 91). These skills are crucial for all grade levels, whether elementary, middle, or high school (Israel et al., 2015). Moreover, as further evidence of the benefit of coding curricula, Litts et al.'s (2017) study demonstrated students could better read code to control a circuit and draw a working circuit after being instructed on how to do so. Such findings provide testimony to coding instruction's effectiveness in developing a variety of skills that are advantageous in all grade levels across disciplines.

Parents from a wide segment of communities and educational leaders regard coding as a way to develop creativity skills within students (Project Tomorrow, 2018). Kim (2019) defined creativity as "the ability to define and solve problems and recognize new possibilities and opportunities. It is a core competence involving flexibility to deal with greater diversity and uncertainty in future society" (p. 68). Students also predominantly agreed with the sentiment above, as 58% of students in Grades 6–8 believed coding engenders creativity, and 51% of high schoolers identify creativity as the most important rationale to offering coding instruction (Project Tomorrow, 2018). As an example of creativity, students in the Teen Design program at the Philadelphia Public Library created apps dealing with bullying, among other issues they encountered (Baker-Doyle, 2018). Additionally, the HIVE Network in Chicago, which encourages STEM development in youth, assisted youth and adults in developing RideW/Me, an app that helped students find means of travel to extracurricular activities because transportation costs were at times prohibitive (Baker-Doyle, 2018). These opportunities inculcated a powerful sense of leadership and civic activism within students (Baker-Doyle, 2018) and highlighted the creative potential when coding.

Occupational

Students' understanding of coding intricacies and acquisition of CT skills will better enable them to earn well-paying positions in a variety of domains in the future and consequently satisfy the current demand for candidates with such knowledge. Indeed, “the need for programming professionals is still growing. The European Commission estimates there are 700,000 unfilled vacancies for IT practitioners, of which programmers outnumber other IT professionals by ratio 5 to 1” (DePryck, 2016, p. 27). Coding is deemed as an essential skill, especially for future occupations (Tuomi et al., 2017; Wilson & Moffat, 2010). Asunda (2012) stated “an increase in STEM literacy will very likely result in a workforce that is capable of assuming technical occupations in a knowledge-based society” (p. 54) and thus remedy the ills pronounced by the European Commission. These contentions were highlighted by Farooq et al. (2012) with their assertion that a:

> basic understanding and knowledge of computer programming is . . . desirable for . . . researchers from all domains. Above all, it is increasingly becoming evident that a basic understanding of computer programming is a need for . . . people belonging to various different domains of life. (p. 136)

Furthermore, Hutchison et al. (2015) offered a corroboratory statement:

> Coding introduces students to the disciplinary literacies used by computer engineers, game designers, graphic designers, and more. Providing students with exposure to coding apps simultaneously provides them with the opportunity to learn specialized language and exposes them to the types of reading and writing performed in professions involving computer programming of any kind. (p. 495)

As one example, Scratch cultivates a wide spectrum of capacities. Hutchison et al. (2015) affirmed

> Scratch can be used as a tool for students to explore what it is like to be a computer programmer, game designer, digital artist, illustrator, writer, and so on. Students will simultaneously have the opportunity to discover the disciplinary skills and literacies involved in such careers and practices and use their literacy skills to create and compose digital products. (p. 497)

Rich et al. (2017) underscored the voices of educators and their rationales encouraging the integration of coding in the curriculum, which included preparing students for various occupation sectors in the future and affording them opportunities in the classroom to decide whether it is something worthwhile to pursue. These sentiments are in line with a Google (2015) report indicating teachers' and principals' beliefs that CS skills are applicable in a wide segment of occupations. Such perceptions from educators and their underlying reasons serve as potent reasons to expose students to coding.

Wu et al.'s (2020) study demonstrated most teachers from different countries believed coding skills are necessary for occupations even beyond programming. However, schools have inadequately prepared students for future employment (McCoy-Parker et al., 2017). Soykan and Kanbul (2018) simply yet potently stated, "By starting coding education at an early age, students will establish a solid foundation for any 21st-century career" (p. 183). Indeed, McCoy-Parker et al. (2017) cited the World Economic Forum's suggestion that 65% of elementary school children will find careers that do not yet to exist. According to the U.S. Bureau of Labor Statistics, the demand for positions in information technology will grow by 12.5% from 2014 to 2024 (Bers, 2018). Parents' perceptions reflect this trend as well. Project Tomorrow (2018) highlighted that parental belief regarding coding's impact on developing workplace skills rose from 28% in 2014 to 45% to 2018. In the United States, “reports indicate that the country will continue to suffer a serious shortage of software professionals for the foreseeable future, and in many other fields familiarity with computer programming and data manipulation will be increasingly in demand” (King, 2015, pp. 21–22). Rich et al. (2020) augmented this point as well, stating the supply of skilled workers cannot match the demand and as such, coding instruction has been, to a greater degree, offered in K–6 schools. Cutumisu and Guo (2019) additionally suggested STEM careers are linked to its exposure in the middle and high school levels. These assessments accentuate the significance of offering coding instruction and developing CT skills in the classrooms. As a result of all this evidence, more can be done in our classrooms and something so simple as exposing students to coding applications can help them think like scientists, mathematicians, or engineers (Estapa et al., 2018), which will subsequently aid them in meeting the challenges of the 21st century workplace (CODE.org, n.d.-b).

Wing (2006) asserted CT has impacted various disciplines such as statistics, biology, and economics. For instance, statistics departments are hiring computer scientists because the field has started to employ machine learning to assess data size that was once impossible to fathom (Wing, 2006). Additionally, biology is using computational methods to better understand the structure and functions of proteins (Wing, 2006). Economics has also been influenced by CT as a new field of computational microeconomics has been generated, which focuses on areas such as advertisement placement and online auctions (Wing, 2008). CT even stretches to fields such as the humanities and the arts, as data mining is used to navigate through digital libraries of books and artifacts to better understand humankind (Wing, 2008). According to Wing (2006), “One can major in computer science and go on to a career in medicine, law, business, politics, any type of science or engineering, and even the arts" (p. 35). Although coding or programming is a much narrower discipline than the broader field of CS, it can regardless assist a prospective employee in securing a position from a wide array of fields. CODE.org (n.d.-b) buttressed this point, assessing "the breadth of industries requiring computing professionals is diverse—two-thirds of computing jobs are in sectors other than information technology, including manufacturing, defense, health care, finance, and government" (p. 2). These various reports and assertions accentuate the value of coding instruction, development of CT skills in our classrooms, and their connection to priming students for not only modern but future demands of workplaces.

Social Justice

There has been a gap between the underrepresented, such as ethnic minorities and socioeconomically disadvantaged populations, in the STEM fields (Blustein et al., 2013). Calabrese-Barton and Tan (2018) espoused the following position:

> For many youth, gaining access to STEM is an uphill battle. Inequality and underrepresentation of youth of color and from low-income communities in STEM persist. For such youth pathways into STEM and STEM empowered lives remain filled with obstacles, from access to quality STEM learning experiences to opportunities to engage with STEM in ways that matter in one's life. (p. 763)

Furthermore, the Code.org Advocacy Coalition (2019) asserted female and underrepresented minority students lack access to CS courses. Wang and Moghadam (2017) stated, "While Blacks and Hispanics make up 13.3% and 17.6% of the U.S. population, they are underrepresented in CS from high school to career" (p. 615). As an additional confirmatory point, Wang et al. (2016)

stated Hispanic students reported lower opportunity or exposure to computers at home and at school, and Black students indicated less access to CS at school.

Additional data evincing these contentions indicate:

> With regards to exposure to technology, Hispanic students are least likely to have exposure to computers at home and at school compared to Black and White students. Only 75% of Hispanic students report having access to a computer at home, compared to 85% of Black students and 98% of White students. At school, only 31% of Hispanic students report using a computer every day, compared to 45% of Black students and 42% of White students. (Wang et al., 2016, p. 648)

Yet, Wang et al.'s (2016) study, which included teachers, principals, and superintendents indicated their perception that CS is important and widely applicable. In fact, most of them perceive CS to be at least as important as other core courses such as science, math, and history (Wang et al., 2016). Furthermore, teachers in schools with a larger percentage of students who qualify for free or reduced lunch even perceive CS to be more important to future success than other elective courses (Google, 2015).

Careers in these fields are typically well-compensated and provide upward mobility (Blustein et al., 2013), which makes the inequity even more lamentable. However, educational leaders understand the importance of effecting positive change to afford greater opportunities to all. Project Tomorrow (2018) affirmed this notion:

> A majority of district administrators nationwide (52%) identify closing the achievement gap as a top priority for their schools. When asked to identify best approaches for addressing the achievement gap, district leaders point to the integration of college and

> career-ready skill development within the everyday curriculum. Holding that position, it makes sense that district leaders also value classroom coding experiences. (p. 4)

In Wang et al.'s (2016) study, 56% of teachers suggested CS should be a requirement in the curriculum; however, less than 30% of teachers perceived it to be a priority for their respective schools or districts. States can do more to this effect by providing CS learning opportunities for elementary and middle school students (Code.org Advocacy Coalition, 2019). For instance, studies have shown exposing female students to CS not only impacts their course selections in high school, but it cultivates an equal degree of confidence in CS within girls as boys (Code.org Advocacy Coalition, 2019). According to the Code.org Advocacy Coalition (2019), "If unaddressed, we will continue to exclude entire populations from this fast-growing field and miss out on the innovations and contributions that diversity promotes" (p. 12).

The literature indicates bridging this divide is essential and it can be accomplished by offering coding instruction to all students, so they are better prepared for workplace demands. Providing such opportunities for digital participation to marginalized groups also strengthens their sense of empowerment (Hagge, 2018). Project Tomorrow (2018) referenced Dr. Farber from the University of Northern Colorado, stating:

> This discussion is about equity of learning opportunities for all students. Access to creative coding learning experiences that help students develop important college and career ready skills should be the right of every child today. It is imperative that our nation's schools embrace this new mindset about learning and empower coding for all students across the school curriculum. (p. 8)

The University of California C-STEM Mathematics Center trains teachers to integrate STEM focused approaches to enhance learning in math for students from marginalized populations who struggle with traditional forms of pedagogy (Kuchey & Flick, 2017). Cheng, the director of the C-STEM curriculum and professor in the Department of Mechanical and Aerospace Engineering at UC Davis, stated, "Teaching math with computer programming—either as part of a standard math course or as an elective—can give mathematical concepts context and relevance while still requiring the same amount of rigor as traditional mathematics instruction" (Kuchey & Flick, 2017, p. 50). Cheng affirmed such exercises in programming develops logical thinking and problem-solving abilities (Kuchey & Flick, 2017). More than 200 schools in California have adopted this curriculum with promising degrees of success (Kuchey & Flick, 2017). In fact, a school primarily composed of socioeconomically disadvantaged students achieved a 94% passage rate on the mathematics examination after the integration of the C-STEM curriculum as opposed to a 61% rate before inclusion of the curriculum (Kuchey & Frick, 2017). Such testimony further demonstrates the potential coding has on student achievement; however, the key lies in affording opportunity. Without opportunity, especially for students who are underrepresented in such fields, the gap in student achievement and STEM representation will persist.

Maker activities serve as another opportunity to embed coding and can also serve as a source of empowerment. Calabrese-Barton and Tan (2018) conducted a study and demonstrated the powerful potential for change with maker activities:

> Through their making, the youth identified and responded to problems that affected them but were also deeply linked to their community's unique history and context. For

> example, youth noted a desire for improving access to books and toys, making and sharing how-to videos made "by us for us," and designing fashionable clothing with unique and functional features. The youth imbued their making projects with wisdom and hope in ways that drew upon their community insider knowledge and experience. Youth shared hopes that their projects might help "kids make friends" (e.g., lightup football), "have fun and be less stressed" (e.g., fidget spinner), or "play with scooters outdoors in the late afternoon or evening when it is dark" (e.g., light-up scooter). It also mattered to the youth that responding to these concerns in their making showed others that they "care" (e.g., Phantom jacket) and wish to "help people in our community" (e.g., light-up umbrella). (p. 780)

Such opportunities allowed these students to accentuate the injustices they experience, highlight the rich culture of their communities, and exemplify the lives they lead as children (Calabrese-Barton & Tan, 2018). These steps will inevitably lead to at least a modicum of increased participation among underrepresented groups in fields such as technology and science, but, also, as aforementioned, in a myriad of industries that seek candidates who can fulfill positions based on such skills. Project Tomorrow (2018) affirmed, "Within large school districts with student populations over 25,000, 51% of district leaders say coding is the best way for students to develop workplace skills" (p. 4). With such opportunities, underrepresented groups will have an assortment of vocational avenues available for them to pursue.

Estapa et al. (2018) cited the U.S. Department of Education in highlighting the 2016 National Education Technology Plan's (U.S. Department of Education, 2016) call to teachers to consider equity in the classroom as they use technology, because some students are active learners, and others are passive learners. Use of coding apps is a valuable approach to involve all students, regardless of form of learning, in an active educational environment (Estapa et al., 2018). King (2015) cited the University of Idaho in asserting female students are generally interested in STEM fields at early ages; however, this interest dissipates later in their academic life. Duncan et al. (2014) cited Margolis and Fisher (2003) in stating exposing female students to programming in middle school is important. Coding apps can serve as the engine for engagement and simply show it can be exciting and fun (King, 2015). In fact, King's (2015) research with the software TouchDevelop recorded a high degree of female student engagement, motivation, and enthusiasm, along with positive learning outcomes as they created applications and games. Serafini (2011) noted a similar outcome in his study, specifying female students achieved the same programming outcomes as male students. This finding is unsurprising, as Hutchison et al. (2015) affirmed that, in 2013, more female students engaged in CS through the CODE.org's Hour of Code than at any point in the previous 70 years.

Female presence in computing and mathematics was substantial in 1960 equating to 27%, and this figure expanded to 35% by 1990 (Thompson, 2019). In 1967, *Cosmopolitan Magazine* published an article dubbed "The Computer Girls," which underscored the continued growth and presence of females in programming (Thompson, 2019). In fact, it is contended that the first coder was indeed female, whose name was Lady Ada Lovelace (Thompson, 2019). Nevertheless, female representation gradually diminished and reached 26% by 2013, which signified less representation relative to 1960 (Thompson, 2019). This was partly attributed to the invention and expansion of personal home computers, which reconfigured the students who demonstrated interest in the field (Thompson, 2019). Parents encouraged the use of computers for males, as opposed to females, and mothers were typically less engaged with computers, which adversely impacted interest among young females (Thompson, 2019; Wang & Moghadam, 2017). Furthermore, this general mindset was reinforced in schools (Thompson, 2019), which was evident in Wang et al.'s (2016) study indicating 63.4% of teachers believed boys held greater interest in CS than girls, and 35.4% of teachers suggested boys would be more successful than girls in the discipline. These subtle yet impactful factors snowballed into the creation and maintenance of stereotypes, such as female lack of ability (Thompson, 2019), and underrepresentation of female students in computing (Wang & Moghadam, 2017). Such stereotypes were borne in actual perceptions, as indicated above in the Wang et al. (2016) study. Wang and Moghadam's study asserted additional credence to these contentions. Among all Computer Science A Advanced Placement test takers, 21.9% were female, and compared to the almost 65% overall average passage rate, females succeeded slightly less at 60.5% (Wang & Moghadam, 2017). CS degrees are also conferred to female students at a much lower clip of 18%, and, relative to their representation in the general workforce, female presence in such industries is drastically lower (Wang & Moghadam, 2017).

This historical and even current day context simply reaffirm the necessity to provide female students equitable opportunities and encouragement so they may bridge the gap in STEM representation in school and in the workplace. As case in point, Carnegie Mellon reformulated their admissions process to the CS program by expanding opportunities for students with less experience (Thompson, 2019). This significantly altered the landscape of incoming students, increasing the percentage of female students in the program from 7% to 42% (Thompson, 2019). Moreover, such alterations led to equal graduation rates between female and male students

(Thompson, 2019). In 2006, Harvey Mudd tinkered with its introductory CS course, creating a separate track for novice students, and by 2018, 54% of students who graduated from the CS program were female (Thompson, 2019).

Project Tomorrow (2018) lent additional credence to the need to dismantle false narratives and stereotypes regarding who can and should code, suggesting it must be infused in the curriculum. Mason and Rich (2020) stated, by infusing coding in the curriculum, the perception that programming is better served for male students, which can diminish female sense of self-efficacy, can be altogether avoided and simultaneously ameliorate the gender gap in CS. This point is especially significant given the gender gap in CS. Gunbatar and Karalar (2018) reported there is a gap in self-efficacy and attitude toward programming between boys and girls, which can impact future career choices. Gunbatar and Karalar affirmed such perceptions can be altered by providing visual programming opportunities to girls. In fact, their study demonstrated, after using mBlock programming, there was no difference between male and female student self-efficacy. Other similar studies using Scratch have underscored the same line of reasoning (Gunbatar & Karalar, 2018). Ultimately, "if a high self-efficacy perception and positive attitude towards programming among girls at middle school level can be developed, it can be ensured that girls may make career choices in the field of computer science" (Gunbatar & Karalar, 2018, p. 926). Duncan et al. (2014) supported this rationale, asserting programming in middle school can bolster female students' confidence in high school CS courses. In fact, "positive school experiences with computing curricula increase the likelihood that female students will study CS in the future" (Duncan et al., 2014, p. 64). As a furthering point, career aspirations start materializing before high school (Duncan et al., 2014), which undergirds the importance of offering coding courses to female students earlier in their formative years. As another means of practice, students can be encouraged to participate in STEM activities on websites like Black Girls Code (https://www.blackgirlscode.com) or Girls Who Code (https://girlswhocode.com) to promote involvement of students who have historically been underrepresented in these fields (Estapa et al., 2018).

In addition to female students' involvement, it is vital to focus upon minority groups' interest and immersion in STEM fields. In 2018, according to the Bureau of Labor Statistics, Black employees only comprised 8.4% of all computer and mathematical oriented occupations (Thompson, 2019). Additionally, Latinos accounted only 7.4% of all such positions (Thompson, 2019). In 2017, 20% of Google's technical employees were female, while Black and Latino employees accounted for 1% and 3%, respectively (Thompson, 2019). A Google and Gallup poll from 2016 provided context to these statistics, stating ethnic minorities experience social and structural barriers to accessing and pursuing CS (Lane et al., 2019). Computer Science A Advanced Placement classes typically see minimal enrollment; however, statistics suggest the numbers of Black and Hispanic enrollees are significantly lower (Wang & Moghadam, 2017). In fact, among the total population of Computer Science A Advanced Placement test takers, only 3.9% were Black and 9.2% were Hispanic in 2015 (Wang & Moghadam, 2017). Additional data continue to paint a bleak picture. The overall passage rate for this test was 64.4%; however, it was 36.8% for Black students and 39.3% for Hispanic students (Wang & Moghadam, 2017). This diminished success rate manifested itself in higher education as well as suggested by an 11.4% of CS degree conferral to Black students and 8.5% for Hispanic students (Wang & Moghadam, 2017).

Furthermore, McCoy et al. (2017) referenced Johnson-Bailey and Cervero (2004) in highlighting the inequities experienced by students of color at the hands of faculty members within a predominantly White institution. Minority students in STEM disciplines at that institution reported lack of engagement among professors and felt they tried to dissuade them from continuing in their respective fields (McCoy et al., 2017). Furthermore, many students stated they did not have sufficient support to complete their academic degrees and faculty members were difficult to reach (McCoy et al., 2017). In fact, McCoy et al. (2017) stated:

> They felt unsupported and even weeded out of STEM disciplines in overt (e.g., faculty telling students they would fail or to change programs) and subtle (e.g., faculty who did not allow students to work in research labs or not showing up for meetings) ways. (p. 667)

Students of Color did not feel they acquired the social capital necessary to pursue STEM graduate programs or careers (McCoy et al., 2017). In harmony with this point, Weissmann et al. (2019) lamented the lack of diversity in student and faculty in higher education in the STEM fields. Moreover, underrepresented minorities are generally not attracted to STEM fields and those who pursue these disciplines feel like outsiders (Weissmann et al., 2019).

Lane et al. (2019) believed diversity in the workplace will generate innovative and unique ideas that will translate into the creation of new products and serve as a significant step toward social justice. Although financial programs and recruitment efforts made some headway in countering these systemic issues, a profound gap nevertheless remains (Weissmann et al., 2019). Lane et al. (2019) attributed this gap to inadequate student recruitment of minority groups, lack of diversity in faculty, and discriminatory behavior in the workplace and in the educational arena. One potential avenue to enhance achievement among diverse groups is to engage in paired programming (Lane et al., 2019), which may result in increased interest in forthcoming grade levels and even as a career avenue. Nevertheless, the aforementioned evidence potently demonstrates the importance of offering coding instruction to all students from a young age and the absolute necessity to overcome any barriers, institutional or otherwise, and evince that all students, regardless of gender or ethnic background, have the desire and ability to succeed in any discipline, including in STEM. In fact, Tissenbaum and Ottenbreit-Leftwich (2020) averred "knowing how to code is a critical factor in women and underrepresented students succeeding and persisting in post-secondary CS education. However, we need to ensure they have relevant experiences at the K–12 levels first" (p. 42). Tran (2019) conducted a study that effectuates this suggestion. Academic programs from two different school districts embedded CT for underrepresented populations including socioeconomically disadvantaged students, Latino students, and English-Language Learners. The program heightened interest in CS and enhanced CT and problem-solving skills (Tran, 2019). According to Tran (2019), "Participants made crucial connections to out-of-school activities including on-demand job-related soft-skills such as communication and teamwork, highlighting benefits gained from the opportunity to learn coding" (p. 22). Tran (2019) also asserted:

> Most interesting from the study was the innate ability of students to persevere as they engaged in collaborative work rather than expecting immediate success when confronted with challenging tasks. The learning experiences made an indelible impact on participants as they embraced collaborative challenges to persist mentally and academically. Increased computational competencies primed early interest for participants to engage in

future CS studies, thus, promoting positive perceptions as they enter secondary school or beyond. (p. 22)

Such testimony cogently demonstrates all students, regardless of socioeconomic background, will succeed if they are simply given the opportunity to do so. Of additional significance was students' ability to connect these learnings to curriculum disciplines (Tran, 2019), which highlights the general the ability of coding to positively impact learning in a variety of subjects and better prepare students for occupational life. To be sure, states have taken impactful steps to counter the traditional gap in representation, yet even the states that have made the greatest strides are a long way from gender or racial parity. Advocates for CS education must ensure policy initiatives are centered on principles of equity and diversity and states must revisit policies often to ensure they lead to equitable and diverse outcomes (Code.org Advocacy Coalition, 2019).

Integrating coding instruction in the classroom presents several challenges, including a lack of technology and a paucity of prepared teachers; however, allocating sufficient resources to offer this opportunity to students will yield immeasurable benefits, including improved student achievement, enhanced preparation for 21st century occupations, and greater access to STEM academics and occupations for marginalized populations. A varied and intuitive range of coding resources are available to teach students from kindergarten until the end of secondary school. Doing so will fortify the STEM pipeline at the higher education level, which will in turn diversify the pool of applicants and employees of STEM related occupations.

Conclusion

Chapter 2 identified the focusing lens of this dissertation, it explained its foundational root, and explained how constructionism can be pragmatically applied in the classroom. The literature review that proceeded underscored relevant concepts to coding instruction, including CT. It furthermore highlighted the three-pronged set of benefits emanating from coding's instruction in the classroom, including enriched student achievement, enhanced occupational preparedness because of developing multifaceted skills, and diversified STEM representation in academics and in workspaces. The final piece of the literature review described the potential avenues for coding's implementation in the classroom and the challenges that can result from it.

CHAPTER 3

METHODS

This chapter focused on the methodology and analytical plan of the study. Important components such as the research questions, participants, and instrumentation are expounded upon to provide the reader with the rationale for the study's foundation. The survey is a point of emphasis in the chapter as it was the sole data gathering tool. As such, the preexisting surveys adapted to create my instrumentation are acknowledged, along with identification of some specific questions to provide the reader a firmer sense of the scope and breadth of the survey.

Research Questions

1. What is K–12 ADLA Catholic school teachers', administrators', and STEM directors'/technology coordinators' understanding of coding?
2. What is their perception on its instruction's impact on students' academic achievement, occupational preparedness, and on its ability to enhance STEM representation in school and in the workforce?
3. How do K–12 ADLA Catholic teachers', administrators', and STEM directors'/coordinators' epistemology and pedagogy impact their inclination to integrate the tenets of constructionism and coding?

Methodology

As the principal investigator, I had the responsibility to conduct the study in a diligent and ethical manner to safeguard the individuals who participate in the study. As such, this document references the Experimental Subject Bill of Rights in Appendix H. Furthermore, during my studies at LMU, I participated in and completed sessions as part of the Collaborative

Institutional Training Initiative (CITI) to prepare me to conduct the study in a principled manner. The certification is referenced in Appendix C, along with my chair professor Dr. Elizabeth Reilly's National Institute of Health (NIH) certification, as noted in Appendix D. With those invaluable lessons in mind and with the guiding assistance of Dr. Reilly, I designed the survey instrumentation.

I used a survey-based quantitative methodology to describe teachers', administrators', and STEM directors'/technology coordinators' understanding of coding entails, assess their perceptions regarding its potential impact on academic achievement, its value to occupational preparedness, and capacity to bridge the disparity in STEM fields, and explain if a link exists between the respondents' epistemology and pedagogy and their inclination to incorporate the tenets of constructionism and coding instruction in the classroom.

Descriptive statistics, in the form of means, modes, standard deviations, and percentages, were used to analyze the data. These measures were employed to not only gauge the perceptions of the entire population of respondents, but additional analyses were conducted on various sets of subgroups as well. The subgroups' data were isolated and used to compare their perceptions on the survey items that were at the crux of my three research questions. I wanted to assess whether there was any difference in the subgroups' perceptions because each group holds a unique space in the educational environment. These potential differences would bear importance on the implications and recommendations of my study.

The total sample size for the survey was 191. The first set of subgroups was evaluated based on their responses to Question 65, which asked to identify their role at school. The choices were teacher, administrator, STEM director/technology coordinator, and multiple roles. The data yielded 108 teachers, 38 administrators, and 38 participants who held multiple roles such as teacher and administrator. Five individuals did not respond; therefore, they were not placed in any subgroup and thus were excluded from this part of the analysis. However, they were included in the overall assessment when the measures were calculated for the entire population of respondents. Moreover, there were two individuals who identified themselves as STEM personnel. However, due to the small size, I did not include this as a separate subgroup, and they were excluded from this subgroup analysis as well. Identical to the five individuals who did specify their role in school, these two individuals' responses were part of the descriptive statistics for the entire pool of participants.

Additional subgroups' data that I isolated and evaluated was based on gender. 141 respondents were female, 43 were male, five participants did not respond to this question, and two chose the decline to state option. Data from these seven respondents were not evaluated in this set of subgroups. No respondent chose the nonbinary or other option.

An additional set of data that was isolated regarded level of instruction (elementary, middle, and high school, along with teachers who instruct in multiple grade clusters). The data were based on participants' responses for Question 66, which asked participants which grade level they instructed in. There was also an option for those who do not teach, such as administrators. Participants were allowed to choose more than one response choice. Fifty-five participants were from the elementary level, 38 were in the middle school level, 39 taught high school, and 27 instructed in more than one cluster. These 27 individuals were placed in their own subgroup. The administrators were not included in this subgroup analysis because they simply did not teach in the classroom and because this subgroup was assessed in the teacher, administrator, and multiple role set of subgroups and in a more expansive analysis of subgroups discussed below. Additionally, five people did not respond to this question; therefore, they were omitted from this set of analyses.

In addition to a descriptive assessment of the aforementioned subgroups, descriptive and inferential analyses were conducted with a software called Statistical Package for the Social Science (SPSS; IBM, 2022). Its security and privacy statements are included in Appendix N. Data from these subgroups was initially organized using Microsoft Excel, which included information such as participants' response identification numbers from Qualtrics (2022), the category of subgroups to which they belonged, their response choices for every question pertaining to the constructs, and the means of participants across each construct. This file was subsequently imported into SPSS. The subgroups that were analyzed included elementary, middle, and high school teachers, teachers who instruct in multiple grade clusters, and administrators. The data yielded 54 elementary teachers, 35 middle school instructors, 34 high school pedagogues, 25 individuals who taught in multiple levels, and 38 administrators. Although this discrepancy will be further elaborated upon in Chapter 4, it is important to provide a brief summation of the inconsistency in responses. In Question 65, which asked participants about their role in school and allowed participants to choose more than one answer choice, 38 individuals stated they were administrators. However, in Question 66, which asked participants about their grade cluster of instruction, such as kindergarten to second grade and third through fifth grades, the number of participants who chose the option that identified that they do not teach, such as administrators, were 27. I decided to use the administrators' responses from Question 65 and all other participants' responses in Question 66 to create subgroups for my final set of analysis. This decision was based on my trust in the original responses by administrators in Question 65. Additionally, five participants did not respond to this question and were consequently excluded from this specific evaluation. This analysis was the most expansive relative to the other assessments and it was formulated differently. This study was based upon three research questions and five constructs were created to underscore the crux of each inquiry. Research question one had one construct, the second inquiry had three constructs, and the final research question had one construct. At least three items were included in each construct to reduce measurement error.

The first construct, comprised of five items, highlighted participants' understanding of coding. It included items evaluating their general and more nuanced understanding of coding, their perceived effectiveness at instructing block- and syntax-based coding, and their ability to integrate coding in the content curriculum. These last three items were not in the same block as the first two questions in the survey; rather, they were in a block titled implementation that was created to gauge the possibility to establishing coding in the classroom. However, because integrating questions from different blocks into a construct were not problematic and it assisted in gauging participants' understanding of coding, these items were combined to establish a construct. The ability to instruct block- and syntax-based coding is part and parcel to understanding coding.

The second research question was multilayered, focusing on participants' perceptions regarding coding's impact on student achievement, occupational preparedness, and diversifying STEM representation. Consequently, three separate constructs were created for this question. Items for the second construct, which consisted of five items, included whether coding develops a student's logical thinking ability, problem-solving, collaboration skills, performance in different subjects, and engagement in school. Three inquiries encompassed the third construct and probed perceptions related to coding's impact on students' ability to obtain positions in technology specific fields, in sectors beyond technology, and whether coding is an important 21st century literacy. The item on literacy was specifically chosen because it relates to coding's connection to an evolving domestic and international economy that is centered upon technology. The fourth construct, composed of four items, emphasized coding's impact on STEM representation, and it queried participants on coding's ability to bridge the divide in K–12 STEM interest between males and females and different ethnic groups along with its instruction's potential to diminish the disparity in STEM occupations held among females and males and between different ethnic groups.

There was an attempt to create a fifth construct highlighting the final research question, which gauges the connection between pedagogy and epistemology and an inclination to integrate constructionism and coding in the classroom. Eight items were included, such as whether direct instruction is an effective means to student learning, if student collaboration and communication are important components of learning, and if learning is perceived to be what the teacher has taught. To assess whether these items were interrelated between each other and with the test, the Cronbach's alpha measure was used. A more elaborate explanation of these measures will be offered in Chapter 4, but it is important to note here that the first four constructs easily surpassed the 0.7 alpha threshold; however, the fifth set did not. Attempts were made to enhance the Cronbach's alpha, including recoding certain questions and omitting others from the assessment; however, such efforts were fruitless. These will be further explained in Chapter 4. Ultimately, a construct was not created, and the data were not assessed with inferential analysis; however, its data were explicated with descriptive measures in Chapters 4 and 5.

Inferential statistics were additionally conducted to augment the vigor of the study and potentially infer generalizations to all others in the respective subgroups within schools in the ADLA. The analysis of variance, or ANOVA was used, and with the *F* ratio, scores from two or more groups were evaluated to determine if there was a statistically significant difference in means at a particular probability level or if the divergence stemmed from sampling error (Mills & Gay, 2019). Simple or one-way ANOVA gauges one variable across two or more groups (Mills & Gay, 2019), which was the form of ANOVA employed in this study.

Furthermore, assumptions such as normality and homogeneity of variance were checked to ensure the interpretation of the results were reliable alpha (P. Ruengvirayudh, personal communication, March 14, 2022). For instance, the *p* value of the Levene test was checked to determine whether the Welch or ANOVA *p* values would need to be assessed alpha (P. Ruengvirayudh, personal communication, March 14, 2022). If the Levene test's *p* value was equal to or below .05, the null hypothesis was rejected, indicating a violation of the assumption of homogeneity and lack of uniformity in variances across the groups alpha (P. Ruengvirayudh, personal communication, March 14, 2022). Furthermore, this would suggest the results were not due to chance or a random occurrence. At this point, the Welch test, which holds an alpha value of close to .05, was measured to determine whether there is a statistically significant difference between the groups. If statistical significance was established at the .05 or below *p* value mark, the post-hoc test of Games-Howell was used to quantify pair-wise differences in means between groups and thus identify the source or sources of the difference in means alpha (P.

Ruengvirayudh, personal communication, March 14, 2022). *P* values equal to or below .05 demonstrated a statistically significant difference in means. Conversely, if the *p* value for the Levene test was above .05, the null hypothesis would not be rejected, and the ANOVA measure would be assessed. The Levene test determines whether the variances between two groups is equal (Mills & Gay, 2019) and if the value is .05 or below, then equality of variances cannot be assumed. If the ANOVA *p* value was above 0.05, then a post-hoc test would not be necessary because there was no statistically significant difference between the groups. However, if the ANOVA measure did demonstrate statistical significance, the Tukey honestly significant difference (HSD) post-hoc test, which controls Type I error effectively, was evaluated to determine the root of the difference in means alpha (P. Ruengvirayudh, personal communication, March 14, 2022). The Tukey HSD is less conservative relative to the Scheffé test and thus more powerful (Mills & Gay, 2019).

The value of the ANOVA is represented by the *F ratio*; those near 1.00 indicated the means were relatively equal, which represented the null hypothesis alpha (P. Ruengvirayudh, personal communication, March 14, 2022). Moreover, this suggested the difference between the groups was random, unsystematic, and not due to the treatment or dependent variable alpha (P. Ruengvirayudh, personal communication, March 14, 2022). This measure, coupled with the *p* value, which evaluated how well the sample data support the null hypothesis, demonstrated whether there was a statistically significant difference between the means. Moreover, the eta-squared point estimate value, which is the percentage of variance that can be explained by the treatment effect (the subgroups), was considered as it highlights the magnitude of the difference alpha (P. Ruengvirayudh, personal communication, March 14, 2022). Although my sample size was more than sufficient, this value was not impacted by the sample. These figures were gauged to offer reliable and proper conclusions.

The foundational concept of the ANOVA is its comparison of the variance between the different groups and its assessment of variance within members of the same subgroup, although researchers are far more interested in the former measure (Mills & Gay, 2019). This method of analysis is advantageous because it compares all subgroups' means with one test, thereby avoiding Type I error alpha (P. Ruengvirayudh, personal communication, March 14, 2022). A Type I error results when an effect is claimed to impact the population groups, but this is not the case alpha (P. Ruengvirayudh, personal communication, March 14, 2022). At the .05 level of statistical significance, there remains a 5% chance that the conclusion is erroneous alpha (P. Molebash, personal communication, March 14, 2022). A *p* value threshold beyond .05 enhances the possibility of reaching an inaccurate conclusion, thus that mark was used. I considered any *p* value below .05 to be a statistically significant difference in means between groups, and thus the null hypothesis was rejected. Based on this standard of analysis and my three research questions, the following null and alternate hypothesis were established:

1. Null Hypothesis: There will be no statistically significant difference in means of perceptions between subgroups.
2. Alternate Hypothesis: There will be a statistically significant difference in means of perceptions between subgroups.

Generally, conducting multiple *t* tests to compare the means of two groups increases the possibility of incurring this error, which is known as test-wise alpha level (P. Ruengvirayudh, personal communication, March 14, 2022). The sum of these test I errors is called experiment-wise alpha (P. Ruengvirayudh, personal communication, March 14, 2022). However, the

ANOVA evaluates all mean differences simultaneously; therefore, it avoids the problem of an inflated experiment-wise alpha (P. Ruengvirayudh, personal communication, March 14, 2022). Therefore, due to this inherent advantage and the existence of five subgroups, I used the ANOVA test to conduct inferential analysis.

When data provide sufficient evidence to reject the null hypothesis, results can be generalized to the entire populations of those subgroup comparisons. Ultimately, through this multilayered inferential process, the sample data became the basis for establishing general conclusions about the populations in my study.

The survey, referenced in Appendix K, was based on self-inspired questions and five pieces of literature, including “Coding in K–8: International Trends in Computing Education with Primary-Aged Children” authored by Rich et al. (2017), “Teacher’s Perceptions and Readiness to Teach Coding Skills: A Comparative Study Between Finland, Mainland China, Singapore, Taiwan, and South Korea” authored by Wu et al. (2020), “Development and Analysis of the Elementary Student Coding Attitudes Survey” written by Mason and Rich (2020), “Measuring Teacher Beliefs About Coding and Computational Thinking” composed by Rich et al. (2020), and “Relational Analysis of Personal Epistemology and Conceptions about Teaching and Learning” authored Chan and Elliot (2004).

The survey was divided into eight blocks, commencing with separate sections on informed consent and instructions to the survey, it continued with individual sections on understanding of coding, its benefits, means to implement it in the classroom, epistemology and demographics, and it concluded with an invitation to submit an email address to be entered into a raffle drawing. Examples of questions from the block on understanding of coding included why students will want to learn coding and how long a coding lesson should be. The section on its instruction's benefits prompted participants to respond to Likert-scale items with choices ranging from *strongly agree* to *strongly disagree*. Participants were asked to respond to items such as "coding develops students' problem-solving abilities" and "coding is an important 21st century literacy." The section on its implementation included Likert-scale items gauging respondents' perceptions on the challenges that can be faced by educators and administrators in implementing coding instruction. The following segment gauged data to determine whether there was a link between epistemology and pedagogy and a penchant to incorporate constructionism and coding in the classroom. Finally, the survey requested demographic information such as participants' age and subject areas of instruction to provide greater context and depth to the study.

Based on the research conducted, no literature was uncovered regarding coding instruction in Catholic schools in the ADLA. As a Catholic school teacher for the past 12 years, I humbly, yet firmly believe coding instruction possesses the potential to enhance student achievement in the classroom, enhance their employment prospects, and diversify the representation in STEM academic and occupational fields. My capacity as a teacher in the ADLA and my zeal to best prepare each of my students to reach their potential inspired me to focus on this issue in the ADLA. I furthermore believe a quantitative study to comprehend the perceptions of a wide range of teachers like myself, administrators, and technology/STEM coordinators across the ADLA will best position me to assert this claim. Indeed, "quantitative research values breadth, statistical descriptions, and generalizability" (Leavy, 2017, p. 87). Within the umbrella of quantitative methodology, an interval measurement scale, particularly in the form Likert-scale items, was used. Respondents primarily chose from a five-point set of response options, from *strongly agree* to *strongly disagree*. These choices were ranked in order and represented equal intervals (Mills & Gay, 2019). The response choices largely included *strongly agree* (1), *somewhat agree* (2), *neither agree nor disagree* (3), *somewhat disagree* (4), and *strongly disagree* (5). The tables in Chapter 4 must be assessed with this important point in mind.

It was my belief that this mode of research would encapsulate the greatest number of respondents and thus shed an expansive light on this issue in Catholic schools. My intent was to serve as a driving force toward a structural change in the Catholic school curriculum by integrating coding instruction and thus better equip all our students with the tools needed to meet future academic and occupational challenges.

This chapter proceeds by detailing the methodology and the theoretical construct that undergirded my study.

Participants and Procedures

Descriptive statistics, via survey research, was used to underscore the responses of teachers, administrators, and STEM/technology coordinators in K–12 Catholic schools in the ADLA. According to Leavy (2017), "Descriptive statistics describe and summarize the data" (p. 111), and this study's purpose was to highlight the data regarding coding. In fact, Mills and Gay (2019) averred, "In some studies, particularly survey studies, the entire data analysis procedure may consist solely of calculating and interpreting descriptive statistics, which enable a researcher to describe many pieces of data meaningfully with a small number of indices" (p. 417). Due to mitigating circumstances, which are discussed in Chapter 5, the number of participants was limited to 191. Although I would have preferred to have a larger pool of respondents, the data I gathered were nevertheless compelling and sufficient to analyze. The breakdown of the pool of participants was provided above in the methodology section.

Inferential statistics were also employed to examine the data in this study. Inferential statistics allows researchers to make computed inferences based on data from a limited number of participants (Mills & Gay, 2019). Such measures do not guarantee or prove assertions, but they instead allow researchers to determine the probability that the data from the sample of participants can be extrapolated or generalized to the entire population of the group (Mills & Gay, 2019). An ANOVA was used to compare the means of multiple groups; if a significant statistical difference was noted, a post hoc test was conducted to determine where the discrepancy resided.

The study was cross sectional because it collected participants' responses at a particular point in time (Leavy, 2017; Mills & Gay, 2019). According to Mills and Gay (2019), "Cross-sectional designs are effective for providing a snapshot of the current behaviors, attitudes, and beliefs in a population" (p. 202). Furthermore, purposeful sampling was employed to identify potential respondents. According to Leavy (2017), "Purposeful sampling is based on the premise that seeking out the best cases for the study produces the best data, and research results are a direct result of the cases sampled" (p. 79). Specifically, due to my various access points as a teacher in a Catholic school and as a student at a Catholic university, purposeful sampling was used to highlight schools within the ADLA in anticipation it would yield a representative picture of Catholic school educators', administrators', and STEM/technology coordinators' understanding and perceptions of coding instruction in the ADLA.

I requested important demographic information from the respondents, which conferred greater depth and context to the study. Understanding vital pieces of information such as age, courses instructed, and number of years in education lends itself to a more vivid portrayal of the overall picture regarding coding instruction in the Catholic school curriculum. Specifics of the data are detailed in Chapter 4; however, the relatively robust responses in favor of coding instruction demonstrate, regardless of respondents' different age groups, racial backgrounds, instructed courses, and years of experience in education, there is firm agreement in coding instruction's benefits and a general willingness to integrate its instruction in the curriculum.

The first question in this segment of the instrumentation queried the respondents' age. Most respondents indicated an age range between 25 and 54. Outliers included teachers aged 20–24 and 55 and above. Most respondents were female with a tally of 141, 43 were male, five did not respond to this question, and two selected the decline to state option. Most respondents had earned a master's degree, and 50 indicated earning a bachelor's degree. Only 14 participants possessed a doctoral degree. Furthermore, most participants were teachers, which equated to 108, 38 were administrators, only two were solely STEM directors or coordinators, 38 people held multiple roles, and five did not respond to this question. There was a relatively even divide between grade-level instruction as 55 respondents indicated they instructed at the elementary level, 38 specified 6–8 instruction, 39 posited they educated at the high school level, and 27 taught in more than one grade-level cluster. Although some respondents instructed subjects such as art and music, the predominant response choices were mathematics, science, history, English, and reading. Almost all the respondents were Christian, which was foreseeable because Catholic schools were surveyed. Moreover, the two principal racial backgrounds were Hispanic or Latino and White (not Hispanic). Thirteen respondents were Filipino, while seven were Black. The preponderance of participants had served at least 6 years in the education profession and although there was almost an equal divergence between credentialed and non-credentialed educators, there were slightly more credentialed educators. Among this group of participants, educators had earned a multiple subject credential relative to a single subject one by a ratio of 2:1.

Survey

I used my access points within the ADLA to procure responses to my survey. Specifically, as referenced in Appendix A, I requested official permission from Dr. Tony Galla, who served as Deputy Superintendent of Elementary Schools, and Dr. Dan O'Connell, who held the same position for all ADLA high schools at the time of the study, to conduct my study. Dr. Galla subsequently provided a list of principals' emails from all ADLA grade schools and high schools. Upon receipt of the list, I emailed the principals explaining my study's focus, its potential benefits for Catholic schools, and inviting them to participate (see Appendix B). I additionally requested the names and email addresses of their faculty and administration so I may compile a database in Qualtrics. Once enough email addresses were gathered, I sent the survey to the potential respondents. The introductory portion of the survey explained its content and purpose and it also included the requisite section on informed consent. Before these steps were taken, the appropriate documentation was sent to and approved by LMU's Institutional Review Board (IRB) to conduct the research. This documentation included the IRB application cover sheet (see Appendix E) and the IRB application questionnaire (see Appendix F).

Instruments

The survey was the sole means to procure data to answer the research questions. The survey began with a statement on informed consent, as established by LMU (see Appendix G). It was critical that respondents understand the nature and purpose of the study and consent to participation. The survey contained fixed -choice questions, including Likert-scale items. Descriptive and inferential data analysis was conducted once sufficient data were obtained from teachers, principals, and STEM personnel from schools across the ADLA.

Validity of a study is based upon whether the data that is gathered informs the results of the study (Mills & Gay, 2019). There are three types of validity: content validity, predictive validity, and construct validity (K. Huchting, personal communication, March 24, 2020). Content validity is based upon the reasonableness of the study and as such, it assesses whether the instrumentation encompasses all the elements thought to be a part of the concept that is being measured (K. Huchting, personal communication, March 24, 2020). My study's intent was to gauge teachers', administrators', and STEM coordinators' understanding of coding, their perceptions of its impact on academic achievement, occupational preparedness, and its ability to diminish the gap in STEM representation in school and in the workplace, and whether there is a link between pedagogy and epistemology and predilection toward constructionism and coding instruction. Although coding is an expansive topic, my study emphasized perceptions on the understanding of coding, benefits to students, and the link between pedagogy and epistemology and constructionism and coding. The research questions and instrumentation were geared to examine perceptions. Furthermore, the study's validity was based on the review of the literature and reliance on preexisting surveys. The preexisting surveys assessed phenomenon such as teachers' perceptions and students' attitudes related to coding instruction. As aforementioned, many of the items were adopted from preexisting surveys, and, because data from the survey were the basis for my recommendations, validity of the study was supported. Moreover, the instrumentation was evaluated by six individuals who adequately offered constructive criticism to enhance its quality. Suggestions were incorporated before the instrument was distributed. Finally, conducting inferential statistics supports the validity of the study because it supports generalizing results of the data to the entire population of ADLA teachers, administrators, and STEM personnel. The scores or the data were the basis for all explanations, recommendations, and generalizations, which lent validity to the study.

Reliability is a measure of the study's assessment of its intended construct and the degree to which the results of the study can be replicated, even by a different principal investigator (Mills & Gay, 2019). There are two types of reliability: temporal and internal (K. Huchting, personal communication, March 24, 2020). Internal reliability is the extent to which each major component of the instrument is interrelated (K. Huchting, personal communication, March 24, 2020). In my survey, other than requesting certain demographic information to provide context to the study, all the other chief blocks of the survey were directly connected to the research questions. Furthermore, internal consistency was assessed by conducting a Cronbach's alpha test. Internal consistency tests such as the Cronbach's alpha assess the extent to which items are related to each other and with the test (Mills & Gay, 2019). Most statistical analyses assume no measurement error exists; therefore, a high reliability coefficient, such as the alpha value of.70 used as this study's threshold for reliability, is critical. Specific items that best represent each research question were selected to create constructs. The strength of each of the Cronbach's alpha values for each set of items, which assessed a different component of the study, will be further discussed in Chapters 4 and 5.

It is my firm belief that coding should be a part of the academic curriculum in Catholic schools and the perceptions of key stakeholders regarding coding's impact on academic achievement, occupational preparedness, advancement of social justice by diversifying the STEM pipeline, and the link between pedagogy and epistemology and constructionism buttressed this contention.

Survey

Survey research is commonly employed in quantitative design (Leavy, 2017) and its standardized questions can be analyzed statistically (Leavy, 2017). The items in a survey gauge people's opinion on an issue (Mills & Gay, 2019). Specifically, I inquired into respondents' understanding of coding; their perceptions of its impact on academic achievement, occupational preparedness, and social justice; and whether educators' pedagogical and epistemological beliefs lend itself to adhering to constructionism and integrating coding instruction; therefore, these five elements served as the operationalized variables. Leavy (2017) posited, "The questions you design around each concept in the study are how you operationalize the variables" (p. 102). Data obtained from the demographic questions connected to the responses on those five variables and thus painted an in-depth picture of the possibility of incorporating coding instruction in Catholic schools.

The survey was conducted via Qualtrics, which is an open-source survey tool (Mills & Gay, 2019), and it contained a section on informed consent, a brief introduction with general instructions (Leavy, 2017), followed by 72 questions. Because Qualtrics was the primary vehicle by which the data were gathered, it was crucial to ensure the data were safeguarded. Appendix I describes the security protocols in place that ensured this. The primary device used to conduct the dissertation's work was additionally protected with an antivirus and internet protection software developed by NortonLifelock and called Norton 360 Deluxe (2022). These measures were crucial to shield all respondents' data from any foreign malicious use.

The survey items were adopted from literature that centered upon coding and modified to an extent to suit the purposes of my study. The survey commenced with a brief description of the purpose of the study and included basic instructions (Mills & Gay, 2019). A Likert scale item, which is an example of a fixed-choice question, was employed in the survey to ascertain subjective data in the form of respondents' attitudes and beliefs (Leavy, 2017). For instance, participants were asked to respond to items such as "What challenges do you think teachers may face with coding instruction in the classroom?" and "What grade levels do you think coding should be instructed?" Additionally, they were asked to reply to statements such as "There is a gap in male and female representation in STEM occupations (i.e., coding/computer programming, information technology)" on a sliding scale of *strongly agree* to *strongly disagree* response options. Questions on epistemology and pedagogy were also presented in the survey to assess its connection to constructionism and coding instruction. Questions to garner objective data (Leavy, 2017), such as participants' age, grade level of instruction, and number of years in education, were placed strategically at the end of the survey to limit respondent fatigue to the questions of greatest primacy (Leavy, 2017). The table of specifications (Appendix L) denotes each category of questions and its corresponding items.

The study's scope was limited to K–12 Catholic schools in the ADLA. Although the response rate was not as robust as preferred, it nevertheless provided ample scope and breadth to generalize the findings to other teachers, administrators, and STEM personnel within the same diocese (Mills & Gay, 2019). The instrumentation can be used to assess the same research questions in other educational domains, including private, public, and charter. The survey was tested by six individuals, who were fellow cohort members from the doctoral program and educators and administrators (Mills & Gay, 2019), and they assisted me in rectifying problem points in the survey.

Data Gathering

Dr. Tony Galla, who served as deputy superintendent of the Department of Catholic Schools (DCS) at the ADLA, assisted me tremendously during my doctoral dissertation process, including providing me the list of principal emails. As of 2021, the DCS did not possess a central database with the contact information of the employed teachers and staff; therefore, Dr. Galla suggested I contact the principals individually. This made it more difficult to acquire the names and email addresses of the faculty; however, during the final week of June 2021, I emailed the principals of Catholic elementary and high schools explaining the purpose of the study, invited their participation, and requested email addresses of their teachers, fellow administrators, and STEM personnel. I requested email addresses in advance to compile a database of contact information in Qualtrics, which was comprised of potential respondents' names and email addresses.

In the days and weeks following the email, I received some responses from principals agreeing to participate, while some declined. Unfortunately, most principals did not respond, which was somewhat expected because the email was sent after the end of the academic year. In the 1st week of August, I initiated another email to principals. This prompted some more positive responses; however, once again, the preponderance of addressees did not respond. At that point, I contacted Dr. Reilly, Dr. Galla, and Dr. O'Connell to discuss what other steps I can take to encourage participation, keeping in mind that this was only the first step in the data gathering process. Even if all the principals responded with their staff members' email addresses, this did not ensure teachers, for instance, would participate in the survey. Upon initiating an email to Dr. O'Connell, I discovered he was no longer employed at DCS, and I proceeded to contact the new point person. This added to challenges I was already experiencing because I had already established contact with Dr. O'Connell, and he had approved my study in the high schools. During my phone conversation with Dr. Galla in the 3rd week of August 2021, I explained the difficulties I had experienced, and he was quick to note that the principals were under immense and unprecedented pressure as they all prepared to embark upon a new academic year amid COVID. He also stated he had not witnessed such challenges except when schools returned to in-person instruction in March of 2021. Suffice to say, the COVID-19 pandemic created numerous obstacles for all schools, and principals were less inclined to participate in such research. Nevertheless, Dr. Galla was generous enough to offer to make personal pleas to the principals during his weekly meetings with various principals after Labor Day in September of 2021. I also sent him a list of principals who had agreed to participate, and he would identify principals who he believed would be willing to participate once I contacted them and mentioned they were referred to me by Dr. Galla.

Ideally, such research would gather data from approximately 40% to 50% of the total population of ADLA teachers and principals; however, given the extraordinary challenges facing this research amid COVID, less data was used. I emailed all the principals on three different occasions in the summer months requesting their faculty's and administrators' names and email addresses. Although most principals did not respond, I did acquire approximately 800 email addresses, which included those of the principals. Some principals preferred to forward the survey directly to their faculty. Once I exhausted this effort and entered potential respondents' names and email addresses in Qualtrics, I distributed the survey on October 8. A reminder email was sent each subsequent week until October 29. Ultimately, after remaining patient to acquire as much data as feasible, I began to analyze the data in early November based on 191 participants' responses. This decision was made after conferring with my chair, Dr. Reilly, and LMU's quantitative methodology professor, Dr. Huchting, on numerous occasions. The data remained sufficient to provide a robust extrapolation of perceptions.

Analytical Plan

Measures of central tendency, comprising of the mean, standard deviation, mode, and frequency (Leavy, 2017), were calculated for the entire group of respondents in the study, which included teachers, administrators, and STEM directors/technology coordinators. A mean is the average of all scores and is considered the ideal measure of central tendency, particularly for interval or ratio data (Mills & Gay, 2019). Likert-scale questions yield interval data because there is an equal distance between each answer choice (K. Huchting, personal communication, January 2022); thus, I calculated the mean to gauge respondents' general perceptions of specific items corresponding to each research question. Additionally, standard deviation is the most frequently employed measure of variability and is generally used with interval and ratio data as well (Mills & Gay, 2019). The standard deviation identifies whether the scores are clustered around the mean or if there are extreme outliers broaden it. This measure assisted me in not only comparing the scores of the items for each research question, but across the research questions as well, and it allowed me to assert detailed and overarching observations. The mode, although not as significant of a measure as the mean or standard deviation, provided an additional small glimpse into the data by identifying the predominant response choice. In descriptive statistics, frequency identifies the number of times a value or response option was chosen (Mills & Gay, 2019). I used percentages as an additional means to explicate the data. These means of analysis enabled me to formulate more nuanced assertions regarding the respondents' perceptions on coding instruction in the curriculum. Demographic information including gender, position in the school, and grade level of instruction provided further invaluable pieces of data after I isolated them into subgroups and analyzed their respective data. This allowed me to better identify perceptions related to coding instruction and how it can be integrated in the Catholic school curriculum.

The research questions gauged the respondents' understanding of coding, perceptions of coding's impact on student academic achievement, occupational preparedness, and coding's ability to diversity representation in the STEM academic and vocational fields, and whether their pedagogical and epistemological philosophy intersects with the tenets of constructionism, thus making the use of technology and coding instruction more likely. The first three aforementioned measures were calculated for all Likert-scale items; however, frequency was computed for all non-Likert scale items. Distinct tables were created for all the Likert-scale and non-Likert scale items corresponding to each research question, which will be discussed in Chapter 4. The validity of the study was supported by a reliance on preexisting surveys regarding coding and on the extant literature. Furthermore, the study's reliability was founded on an instrumentation organized with distinct sets of items responding to each of the research questions; therefore, the measure assessed what was originally intended.

Testing the Instrument

In advance of distributing the survey to the potential participants, the instrument was tested by six people to enhance the validity of the study. A rubric was provided to the testers as well (see Appendix M) for feedback. Tester 1 highlighted concerns such as the overall length of the survey, the confusing language connecting Items 54 and 55, and the possible need to add a demographic question on duration of service in education. Thereafter, I remedied all the noted concerns. Tester 2 also identified survey length as a roadblock to an adequate response rate; however, that was the only identified concern. Originally, the survey reached close to 100 questions; however, I reduced it to 72 questions. Tester 3 did not believe any corrective measures were needed in the survey. Tester 4 did not cite any concerns either. He stated the survey was clear, as exhibited by a consistent logical flow. Furthermore, he suggested the questions were highly appropriate given the target audience, the projected data had a high degree of usefulness, and the items of the survey were varied and relevant to its purpose. Tester five complemented the strength of the survey based on its close connection to the research questions and the purpose of the study. However, he identified logistical issues with some questions, such as the presence of "all of the above" answer choice and the ability to choose multiple answer choices; therefore, he believed providing both options to respondents were unnecessary. Nevertheless, I maintained structure to some of the questions because it is possible, for instance, for a respondent to choose only the first two answer choices. Therefore, for a question with four response choices, all the above would not accurately portray that respondent's beliefs. Furthermore, tester five questioned whether respondents would know what programming languages such as Python and C were as it pertains to Items 61 and 62; however, those questions allow respondents to choose options indicating they are unprepared to instruct such languages.

These recommendations assisted me in crafting a more robust and straightforward survey, which enhanced the potential to obtain meaningful data that represented the respondents' understandings and perceptions regarding coding instruction in the curriculum.

Institutional Review Board

In preparing to conduct the survey in Catholic schools under auspices of the ADLA, I submitted three forms to LMU's IRB to evince that the research would be conducted ethically. First, the application requested basic information such as my and my chair professor's names, which can be referenced in Appendix E. I explained that I would use the survey results to complete my dissertation and might publish and present the content. Additionally, I explained that I hope to shine a light on the need to incorporate coding instruction in the Catholic school curriculum with the data gathered from schools in the ADLA. I also mentioned I would share results of the data with participants and specifically the school where I was employed at the time of this study. Moreover, I explained my hope that the data will inform our pedagogical decisions moving forward to best prepare students in a shifting global society. Secondly, the questionnaire required detailed descriptions of my methodology (see Appendix F). For instance, I expounded upon the research's background, subject recruitment, procedures, risks and benefits to participants, confidentiality of the gathered data, payment to encourage participation, and my personal qualifications and training to engage in the research. Third, the informed consent form (see Appendix G) requested a description of the study's purpose, risks to participants, benefits resulting from the study, incentives offered to potential participants, and steps taken to ensure the confidentiality of the gathered data.

Conclusion

Chapter 3 detailed the methodology and analytical plan of the dissertation, which are undergirded by the theoretical paradigm of constructionism. Constructionism's precepts are at the very core of what I aspire to accomplish in all our classrooms. However, for this study, I used survey instrumentation to support my contention that coding should be a part of the curriculum in Catholic schools in the ADLA. A constructionist classroom amplifies students' role in the learning process and it enhances their motivation and engagement as they apply what they learn in math, science, or reading in novel ways, whether it is by way of constructing a robot, creating an artifact, or producing a story. The opportunities in this type of an educational environment yield endless opportunities for teachers to engage their students and for students to learn. Skills such as creative thinking, problem solving, and collaboration are developed as a result of such activities and these are crucial for all students, let alone students who have typically been marginalized from such opportunities in the classroom. A curriculum that equitably affords all our students these opportunities better positions them to become the future leaders of our societies.

CHAPTER 4

RESULTS/MAIN FINDINGS

The purpose of this study was to gauge teachers', principals', and STEM/technology coordinators' understanding of what coding entails, assess their perceptions on coding instruction's impact on student academic achievement, occupational preparedness, and ability to alleviate the divide in STEM representation in school and in the workplace, and ascertain whether a link exists between their pedagogy and epistemology and predilection to integrate the tenets of constructionism and coding. Due to my capacity as a teacher within the ADLA, my study focused on the educators within this population group. It was intended to highlight the need to incorporate coding instruction in the Catholic school curriculum as an additional avenue to develop various skills such as critical thinking, logical reasoning, and collaboration within students, better prepare them for various occupations that seeks such skills, and further diversify the STEM fields.

I conducted a quantitative study based on a 72-fixed-choice-question survey using Qualtrics. The survey was organized into different blocks that corresponded to each of the research questions, which allowed me to analyze the data with more ease. I compiled a database with all the principals' email addresses within the ADLA, and I proceeded to email each one to explicate the nature and purpose of the study, invite their participation, and request their staff's email addresses. Additional emails were sent as the majority did not respond to my initial request. When principals agreed to participate and provided their staff's names and email addresses, I encoded the information into the contacts database in Qualtrics. Additionally, some principals were willing to participate in the study, but preferred not to provide their staff's contact information and instead requested that I send them the link to the survey which they would then forward to their staff. Upon entering all the information in Qualtrics and determining that there would be no additional principals who were willing to participate based on their lack of response to my multiple attempts at contacting them, I distributed the survey commencing on Friday, October 8, 2021, to all the potential respondents requesting their participation in the survey. A reminder email was sent each subsequent Friday for the next three weeks. The final reminder was sent on October 29. Approximately, one week later after that point, I imported the data from Qualtrics into Microsoft Excel to begin data analysis.

Measures of central tendency, which included the mean, standard deviation, and mode, were calculated for all Likert-scale items. Furthermore, the frequency was calculated for all non-Likert scale items, which included questions prompting participants to identify reasons why their students would and would not want to learn coding. The addition of percentages provided a deeper analysis of the data as well. Furthermore, the analysis of different sets of subgroups based on position in the school, gender, grade level of instruction, and a more expansive compilation of subgroups including teachers in the elementary, middle, and high school level, participants that instruct in multiple grade bands, and administrators yielded a more nuanced picture of the respondents' perceptions.

This chapter will reidentify each research question, provide various tables highlighting descriptive and inferential data, offer an analysis of each table, and conclude with an overarching assessment of the data.

First Research Question

Due to a lack of research on coding instruction in the ADLA, three different lines of query were focused upon. The first question attempted to identify what teachers, administrators, and STEM coordinators understood about coding. Thus, the question asked: What is K–12 ADLA Catholic school teachers', administrators', and STEM directors'/technology coordinators' understanding of coding?

Survey Results

The following section will provide a robust discussion explaining the results of the various analyses conducted as part of the first research question, including, but not limited to, an assessment of the entire set of data, an analysis of data based on the position of the respondents, and an evaluation of responses based on the participants' grade level of instruction.

Analysis of Total Sample Population. As indicated previously, response choices for the Likert-scale items ranged from *strongly agree* (1), *somewhat agree* (2), *neither agree nor disagree* (3), *somewhat disagree* (4), to *strongly disagree* (5). Respondents were queried on a variety of questions to indicate their level of understanding about coding, including whether students will want to learn coding, whether it is valuable to learn coding, and whether there is a gap in male and female interest in STEM subjects and occupations. The mean, standard deviation, and mode scores for the Likert-scale items corresponding to the first research question are highlighted in Table 1.

Table 1

Total Sample Population's Descriptive Statistics for First Research Question's Items

Survey Item	*M*	*SD*	Mode
My students will want to learn coding.	2.07	0.85	2
There is a disparity in access to technology in Catholic schools in the ADLA.	2.14	1.03	2
There are specific ethnic minorities that lack exposure to coding instruction in schools.	1.94	0.98	1
Students who intend to pursue a career related to programming should be the only ones who are taught coding in schools.	4.3	1.07	5
Students who intend to pursue a career related to programming should be the only ones who are taught coding in schools.	4.3	1.07	5
It is valuable to learn coding.	1.72	0.73	1
Male and female students can succeed in coding classes.	1.13	0.41	1
Boys are better at coding than girls.	4.52	0.86	5
Girls are better at coding than girls.	4.38	0.93	5
There is a gap in male and female interest in STEM subjects in K–12 Catholic schools.	2.7	1.09	2
There is a gap in male and female representation in STEM occupations.	1.95	0.96	1
There is a gap in interest in K–12 STEM subjects between different ethnic groups.	2.45	1.1	2
There is a divide in STEM occupation representation between different ethnic groups.	2.01	0.94	1
I have a general understanding of what coding is.	2.07	1.01	2
I have a more detailed understanding of what coding entails (examples: I understand how some coding apps work or I have taken coding courses).	3.31	1.42	4

There was a general belief that students will want to learn coding as evidenced by the 2.07 mean score and a mode of *somewhat agree* (2). The standard deviation of 0.85 demonstrated there was some variance away from the mean. Other items, which were evaluated with a frequency measure, delved into the reasons why students would and would not want to learn coding. These items provided a deeper understanding about students and coding and allowed participants to choose multiple answers. There were 109 responses demonstrating the belief that coding would be fun and exciting, students would want to learn it because of its novelty received 117 responses, collaborative opportunities would interest students was chosen 67 times, and 37 indicated other reasons, which predominantly regarded student awareness of its importance as it relates to career readiness/employment opportunities. Nevertheless, when prompted to identify reasons why students may prefer not to learn how to code, 80 affirmed its difficulty as a dissuading reason, 116 suggested students do not have any prior experience, 58 asserted that they would not be interested, and 18 suggested other reasons including the abstract nature of the subject, student age, a disinterest in math and science which will lead to apathy toward coding, and a strong societal gender bias and inequity in access to technology and digital navigation. These are all valid reasons, but many, if not all, can be eliminated simply by beginning to expose students to coding. Because coding is digital-based and many intuitive resources exist to instruct it, it would not be astonishing to find students immerse themselves in the learning process and grow increasingly motivated to further their knowledge.

Additional data from this block demonstrated coding is valuable to learn as expressed by a mean of 1.72, a mode of *strongly agree* (1), and a standard deviation of 0.73. Respondents also believed there is some disparity in access to technology in ADLA Catholic schools as most chose *somewhat agree* (2). This corresponded to the strong belief that ethnic minority groups have disproportionate access to coding instruction in schools as the predominant response choice was *strongly agree* (1). Consequently, the mean was closer to 1 with a score of 1.94 and the standard deviation was 0.98, suggesting there were not many responses that veered far off from the mean. Respondents also showed relative agreement with the statement that there is a gap in interest in K–12 STEM subject among different ethnic groups as the mode was *strongly agree* (1) and the mean was 2.45 with a standard deviation slightly above 1. This demonstrates the predominant choice was strong agreement; however, there was also variation of responses away from the mean. Furthermore, there was an even stronger sentiment that a divide exists in STEM occupation representation as evidenced by a 2.01 mean, a standard deviation of 0.94, and a mode of *strongly agree* (1).

However, participants did not entirely believe girls are less involved in coding, as most neither *disagreed* nor *agreed* (3) with the statement. The mean was 2.62 and the standard deviation was 0.93. Those who *strongly agreed* or *somewhat agreed* with the statement that girls are less involved were then prompted to choose from fixed choices or offer alternate reasons as to why this is the case. Choices included less interest, societal impact due to the perception that coding is more suitable for boys, both of the first two options, and an option to provide a different rationale. Twelve participants elected the first response, 37 chose the second choice, 33 selected both, and five provided justifications such as less access or availability and no previous exposure to coding. These data, although not definitive, signified some participants believed females are less involved in coding relative to males. Nevertheless, respondents affirmed neither boys or girls are better than the other at coding as evidenced by a mode of *strongly disagree* (5) for both items, mean scores hovering around 4.5, and standard deviation scores of less than 1. These data corroborated the stout perception that boys and girls can succeed in coding classes as suggested by the mean score of 1.13, a mode of *strongly agree* (1) and a standard deviation of 0.41. Furthermore, respondents did not believe coding instruction should be limited to only those students who intend to pursue a career related to it as suggested by a mode of *strongly disagree*. Nevertheless, regarding whether there is a gap in interest in STEM subjects between boys and girls, the mode was *somewhat agree* (2), but the mean of 2.7 was closer to 3, which suggests tepid agreement. Respondents also commonly deemed there is a disparity in STEM occupation representation between males and females as demonstrated by the mean score of below 2 and a mode of *strongly agree* (1). The respondents, in assessing their personal understanding of what coding entails, *somewhat agreed* they generally know what coding entails; however, fewer respondents held a more nuanced level of understanding.

Analysis of Subgroups Based on Position. In addition to a general analysis of the entire population of respondents, I evaluated the perceptions of teachers, administrators, and participants who hold multiple roles as it relates to the most important items for each research question in the survey (see Table 2). The total number of participants was 191, and based on the responses for Question 65, which asked participants whether they were teachers, administrators, STEM director/technology coordinator or if they held multiple roles, there were 108 teachers, 38 administrators, and 38 individuals with multiple roles. Five participants did not respond to this question, and two stated they were a STEM director/technology coordinator; thus, I did not include these seven participants in these analyses. I also did not create a separate subgroup for STEM directors/technology coordinators because there were only two people in that subgroup.

Table 2

Descriptive Statistics on Understanding of Coding (Based on Participants' Position)

	Teacher			Admin			MR		
Survey Item	*M*	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode
Girls are generally less involved in coding compared to boys in school.	2.62	0.82	3	2.45	0.92	2	2.84	1.24	3
There are specific ethnic minorities that lack exposure to coding instruction in schools.	2.07	1.00	1	2.05	1.01	1	1.47	0.73	1
It is valuable to learn coding.	1.77	0.77	1	1.82	0.73	2	1.53	0.60	1
There is a gap in male and female interest in STEM subjects in K–12 Catholic schools.	2.80	1.07	3	2.47	0.83	2	2.74	1.35	2
There is a gap in male and female representation in STEM occupations (i.e, coding or programming)	2.18	1.06	1	1.79	0.74	2	1.53	0.69	1
There is a gap in interest in K–12 STEM subjects between different ethnic groups.	2.55	1.08	3	2.42	1.08	2	2.26	1.22	2
There is a divide in STEM occupation representation between different ethnic groups.	2.17	0.99	3	1.95	0.77	2	1.53	0.69	1
I have a general understanding of what coding is.	2.12	1.03	2	2.24	1.05	2	1.76	0.91	1
I have a more detailed understanding of what coding entails (examples: I have used some coding apps myself, or I have taken coding courses in school)	3.33	1.39	4	3.63	1.26	4	2.92	1.58	1

Note. MR = multiple roles.

There were various questions attempting to gauge respondents' understanding of coding (see Table 2), including whether girls are less involved in coding relative to boys. The mean for the teacher subgroup was 2.62, which is not a strong indication either way because it falls in the middle of *strongly agree* (1) and *strongly disagree* (5). However, this number is surprisingly identical to the mean of the entire survey's population's response for this question. The standard deviation for this item was 0.82 with a mode of 3, which indicates respondents primarily chose *neither agree nor disagree*. The mean for administrators was 2.45, the standard deviation was 0.92, and the mode was *somewhat agree* (2). In comparison to the teachers, administrators were slightly more of the mindset that females are less involved in coding relative to females. In contrast, the mean for the multiple roles group was 2.84, the standard deviation was 1.24, and the mode was *neither agree nor disagree* (3). Although these figures are not significantly different than the other two subgroups, these school personnel tended to disagree with the statement slightly more. Nevertheless, the response choice indicates most participants *neither agreed or disagreed* with the statement. If respondents selected *strongly agree* or *somewhat agree* for that statement, they were prompted to identify a reason why that was the case. Fixed choices including "less interest," "societal impact" stemming from a perception that coding is more suitable for males," both "less interest" and "societal impact," and an option for an alternate response were offered. Among the 48 teachers who responded to this item, 41.67% selected societal impact and 35.40% chose both less interest and societal impact as the primary reason/s why girls are less involved in coding. Among the 21 administrators, 57.14% considered societal impact as the primary impediment. There were 14 participants with multiple roles and 57.14% chose both less interest and societal impact as the causes for reduced involvement. The predominant response choice among all three subgroups was societal impact, which aligns with the prevailing literature indicating the various societal inhibitors for female involvement in coding and STEM.

I also surveyed whether respondents believe specific ethnic minorities lack exposure to coding in schools. Teachers' responses yielded a mean of 2.07, a standard deviation of 1.00, and a mode of 1, which suggest the belief that lack of exposure is a reality in schools. The administrators' responses yielded very similar results with a mean of 2.05, a standard deviation of 1.01, and a mode of 1, which once again indicate the same belief. Those who hold multiple roles were even more emphatic in their belief in this reality as suggested by a mean of 1.47, a standard deviation of 0.73, and a mode of 1. Data from all three subgroups demonstrate relative agreement that specific ethnic groups lack access to coding in schools, which is a firm reason to integrate it in the curriculum.

Respondents were also asked to opine on whether coding is valuable to learn. Teachers suggested it is as indicated by a mean of 1.77, a standard deviation of 0.77, and a mode of 1. The administrators' data yielded a mean of 1.82, a standard deviation of 0.73, and a mode of 2. The mean for administrators was greater by. 05 and the primary response choice was *somewhat agree*; therefore, this subgroup did not believe in coding's value as firmly. The multiple role subgroup's responses were more affirmatory because the mean of 1.53 is closer to *strongly agree*. The mode was *strongly agree,* and the standard deviation was 0.60, suggesting the responses were not far off of the mean.

The literature discussed the potential for success for both males and females in coding. As such, I enquired into this point and the teachers were quite exclamatory in their belief that both males and females can succeed. The mean of 1.16, a small standard deviation of 0.48, and a mode of *strongly agree* evince this point. The administrators also firmly believed in this sentiment as evidenced by a mean of 1.08, a standard deviation of 0.27, and a mode of *strongly* *agree*. The mean and standard deviation for the multiple role group was even less and thus, the overarching sentiment with all three subgroups resolutely signals the belief in the potential for success for both males and females in coding. I additionally asked whether there is a gap in female interest in STEM subjects in K–12 Catholic schools. The responses across all three subgroups were thus quite similar to the replies regarding female involvement in coding. The data from teachers generated a mean of 2.80, a standard deviation of 1.07, and a mode of *neither agree nor disagree*, while the administrators' responses produced a mean of 2.47, a standard deviation of 0.83, and a mode of *somewhat agree*. Similarly, the multiple role subgroup had a mean of 2.74, a standard deviation of 1.35 and a mode of somewhat agree. These responses were not as resolute compared to the ones for questions on female potential for success and the value of coding instruction in the classroom, which suggests participants do not strongly believe there is a gap in interest between females and males. I remained on the identical theme of gender and invited responses to whether there is a gap in representation in STEM occupations. Teachers predominantly chose *strongly agree*, but the mean was closer to *somewhat agree* as evidenced by a mean of 2.18. The standard deviation was 1.06, suggesting some responses were *neither agree nor disagree*. The administrators' mean was 1.79, the standard deviation was 0.74, and the mode was *somewhat agree*. As such, these figures indicate somewhat of a difference in perception between these two groups. Although the modes were identical, the administrators' closer proximity to *strongly agree* and smaller standard deviation suggest this group more so believes there is a gap in male and female representation in STEM occupations. In regard to the multiple role subgroup, their mean of 1.53, mode of *strongly agree*, and standard deviation of 0.68 demonstrate a firmer belief in the statement and a closer alignment with the administrators' perceptions.

In addition to queries regarding male and female students, participants' perceptions regarding ethnic minorities and coding/STEM were evaluated. Respondents were asked to opine on whether a gap exists in K–12 STEM interest between different ethnic minorities. The mean for the teachers' responses was 2.55, the mode was *neither agree nor disagree*, and the standard deviation was 1.08. The administrators' mean was 2.42, the mode was *somewhat agree*, and the standard deviation was 1.08. The mean for participants with multiple roles was 2.26, the mode was *somewhat agree*, and the standard deviation was 1.22. Other than the mode, there was not a significant difference in the means and standard deviations. There was not a strong sentiment toward one extreme or the other on the Likert scale regarding this item. I furthermore assessed whether there is a divide in STEM occupation representation. The teachers' data produced a mean of 2.17, a mode of *neither agree nor disagree*, and a standard deviation of 0.99. Although the predominant response choice was *neither agree nor disagree*, the mean was closer to *somewhat agree*, which suggests some respondents *strongly agreed* and *somewhat agreed* with the statement. Administrators' responses generated a mean of 1.95, a standard deviation of 0.77, and a mode of *somewhat agree*, which indicate a firmer belief in the statement. However, those with multiple roles produced responses indicating the strongest belief in the existence of a gap in STEM occupational representation between different ethnic groups. The mean for this subgroup was 1.52, the mode was *strongly agree*, and the standard deviation was 0.68.

A final set of questions that attempted to determine what the respondents understood about coding first asked whether respondents have a general and then a more nuanced understanding of coding. The mean for teachers regarding the former item was 2.12, the mode was *somewhat agree*, and the standard deviation was 1.03. The administrators' data returned a mean of 2.24, a mode of *somewhat agree*, and a standard deviation of 1.05. These two sets of data were quite similar indicating a relatively strong understanding of what coding generally entails; however, the multiple role group had more robust responses as the mean of 1.76, a mode of *strongly agree*, and a standard deviation of 0.91 suggested a greater understanding of what coding involves. In regard to the more nuanced understanding of coding item, which asked respondents if they know how coding applications function, if they have used coding applications, or have taken coding courses in school, the teacher subgroup's data were less bullish. The mode was *somewhat disagree*, the mean of 3.33 trended closer to *somewhat disagree*, and the standard deviation was a more pronounced 1.39, which indicates some variance from the mean. The administrators' data were similar to those of the teachers as their mode was *somewhat disagree* as well along with a mean of 3.63. The multiple role group's mode was once again *strongly agree*; however, the mean of 2.92 and the standard deviation of 1.58 indicates a wide variety of answers, with many responses trending to both extremes on the Likert scale. It was somewhat surprising that the multiple role subgroup's mode signified that more of them possessed a nuanced understanding of coding, but it did make sense because this group had teachers who instructed technology and CS class. Nevertheless, an overarching picture of the data suggests these subgroups lack a detailed understanding of what coding entails.

Analysis of Subgroups Based on Gender. An additional set of descriptive analyses, as noted in Table 3, was conducted on male and female respondents to assess whether there are any points of convergence and divergence on the most crucial items from the first research question.

Based on a sample population of 191, there were 141 females, 43 males, five did not respond to the question, and two declined to state. Consequently, the data were based on 184 participants.

Table 3

Descriptive Statistics on Understanding of Coding (Based on Gender)

	Female			Male		
Survey Item	*M*	*SD*	Mode	*M*	*SD*	Mode
Girls are generally less involved in coding compared to boys in school.	2.60	1.00	2	2.72	0.73	3
There are specific ethnic minorities that lack exposure to coding instruction in schools.	1.84	0.94	1	2.28	1.05	3
It is valuable to learn coding.	1.72	0.74	1	1.72	0.70	2
Male and female students can succeed in coding classes.	1.13	0.39	1	1.09	0.43	1
There is a gap in male and female interest in STEM subjects in K–12 Catholic schools.	2.75	1.17	2	2.58	0.79	3
There is a gap in male and female representation in STEM occupations.	1.88	0.98	1	2.21	0.86	3
There is a gap in interest in K–12 STEM subjects between different ethnic groups.	2.45	1.16	2	2.49	0.96	3
There is a divide in STEM occupation representation between different ethnic groups.	1.96	0.90	2	2.05	1.00	1
I have a general understanding of what coding is.	2.15	1.06	2	1.79	0.83	2
I have a more detailed understanding of what coding entails.	3.41	1.39	4	2.95	1.50	1

Note. The rest of the last item in this table, for brevity's sake, is included here. This was included in a parenthesis. Examples include: I understand how some coding apps work, I have used some coding apps myself, or I have taken coding courses in school.

The first survey item that I analyzed under research question one asked whether girls are generally less involved in coding compared to boys. The mean for female respondents was 2.60 and the mode was *somewhat agree*, while the male responses netted a slightly elevated mean of 2.72 with a mode of *neither agree nor disagree*. The standard deviation for females was greater as well at 1.00, whereas the male respondents' standard deviation was 0.73. Although there is not a significant difference between their responses, the lower mean and mode of somewhat agree for females may be due to societal factors. The literature stated female interest and participation was adversely impacted by perceptions that CS was more appropriate for males due to their supposed enhanced ability in mathematics. Males, in contrast, may not completely understand and feel the effects of these societal factors, which may be a cause for the difference in responses. Those respondents who selected *strongly* or *somewhat agree* were then prompted to respond to the reasons why that is their perception. Response choices for this item included *less interest* (1), *societal impact* (2), *both less interest and societal factors* (3), and a choice for an alternative response, identified as *other* (4). There were 69 females who responded to this item and 15.94% chose less interest, 42.02% selected societal factors, and 36.95% identified both factors as the reasons for diminished involvement. Only 5.8% chose the last option. The 15 males who responded to this question, quite similarly to females, chose societal factors (46.67%) as the chief contributing factor followed by both less interest and societal factors (40%). These data harmonized with literature explaining the powerful impact of societal perceptions upon the persisting diminished involvement among females in CS and STEM subjects.

The survey also assessed whether participants believed coding was valuable. Both female and male means were consistent at 1.72 and 1.72 along with standard deviations of 0.74 and 0.70; however, the modes were different. Females primarily chose *strongly believe*, but the greatest male response was *somewhat agree*. Once again, similar to the exposure of ethnic minority item, although it is a minor difference, it is interesting to gauge why less males, proportionally speaking, identified strong agreement in coding's value relative to females. Even though there is this slight discrepancy, males resoundingly believe females can succeed in coding. The mean of 1.09, mode of *strongly agree*, and minor standard deviation of 0.43 is a robust set of data and is comparable to the female mean of 1.13, mode of *strongly agree*, and standard deviation of 0.39. The female position that girls are less involved in coding, as was explained above, does not necessarily link with the potential to succeed. Female (and male) respondents steadfastly suggest girls can succeed in coding; however, involvement is a different challenge that needs to be addressed.

To gain a better understanding of any discrepancy between males and females regarding coding instruction, the survey also queried interest in STEM. The prior questions focused on involvement and potential to succeed whereas as this question probed whether there is a gap in male and female interest in STEM subjects. The mean for females was 2.75, the mode was *somewhat agree*, and the standard deviation was 1.17; for males, the mean was 2.58, the mode was *neither agree nor disagree*, and the standard deviation was 0.79. The male responses for interest are quite similar to those for involvement, which may indicate they do not perceive female interest or involvement in STEM to be especially problematic. The female responses for female involvement in coding and if there is a gap in interest in STEM were also analogous, but the basis of their similar responses, which may be different than the reasons for the male responses, are possibly rooted in societal factors inhibiting involvement and interest

A subsequent question prompted a response to whether there is a gap in male and female representation in STEM occupations. The data on female responses yielded a mean of 1.88, a mode of *strongly agree*, and a standard deviation of 0.98, while the male respondents' answers generated a mean of 2.21, a mode of *neither agree nor disagree*, and a standard deviation of 0.86. Curiously, the male responses' data were similar to their data on female involvement and interest. Although the mean for this question was closer to *somewhat agree*, the overall picture with the male responses seem to indicate they do not necessarily believe female interest and involvement in STEM, whether academically or professionally, is problematic. The data from females indicates they did not *strongly agree* there is a lack of female involvement in coding and interest in STEM, but there is a lack of representation in STEM occupations. The reasoning for this difference would need to be investigated further to make a proper assessment; nevertheless, this perception does harmonize with the literature, which explicates how female representation in STEM occupational fields wanes relative to males.

Respondents were also asked to opine on whether there are specific ethnic minorities that lack exposure to coding instruction in schools. The mean for females was 1.84 and 2.28 for males, which indicates a difference in perception. The difference in modes further asserts this point as females primarily selected *strongly agree*, whereas males chiefly chose *neither agree nor disagree*. The female standard deviation of 0.94 and 1.05 for males indicates a modicum of divergence away from the mean. It is interesting to explicate the reason for these data's difference. Although the population number of males is significantly less, it is nevertheless noteworthy and warranting of further investigation to identify the rationale for their perception.

Another item surveyed whether there is a gap in interest in K–12 STEM subjects between different ethnic groups. The mean of 2.45 for females and 2.49 for males suggests a slight state of uncertainty. These figures are essentially between *somewhat agree* and *neither agree nor disagree*; the female mode is *somewhat agree* and the male mode is *neither agree nor disagree*. The female standard deviation is greater at 1.16 relative to the male standard deviation of 0.96, which indicates more of the male responses clustered around the mean. There is more divergence away from the mean with female responses and although their mode is *somewhat agree*, there are participants that chose *somewhat disagree* and even *strongly disagree*. There is not a strong response choice on either extreme of the scale among males and females with this item. However, when asked about a gap in STEM occupation representation between different ethnic groups, the responses did indicate some agreement that a divide exists. The female mean was 1.96 and the male mean was 2.05, but the male mode was *strongly agree* and the female mode was *somewhat agree*. The female standard deviation was 0.90 and the male was 1.00 range. The anomaly with these data more so centers upon the male responses. Their mean was slightly higher than females, but their mode was *strongly agree*. This indicates some male respondents selected *somewhat agree* and *neither agree nor disagree.* Nevertheless, both groups do believe there is an issue with STEM occupation representation.

The final set of questions that attempted to gauge the respondents' understanding of coding urged respondents to measure their general and then their more nuanced or detailed understanding of it. The mean for female participants was 2.15 for the first question, but it escalated to 3.41 for the second question. The modes increased as well from *somewhat agree* to *somewhat disagree,* while the standard deviations rose from 1.06 to 1.39. The male responses increased as well, but their perceived level of understanding was stronger to begin with and the increase in the mean from one question to the next was not as great. The means were 1.79 and 2.95, but quite astoundingly, the modes shifted from *somewhat agree* to *strongly agree*. Based on the data, male respondents possess an enhanced general and more nuanced understanding of what coding entails, which may be due to the difference in involvement, interest, and occupational representation that has been underscored in the literature and in these data.

Analysis of Subgroups Based on Grade Clusters. This analysis was based on data from participants in different grade clusters (see Table 4). Their responses on the most significant questions from each of the three research questions were evaluated. Data were organized between participants in elementary school (kindergarten to fifth grade), middle school (sixth to eighth), and high school (ninth to twelfth). A fourth and final subgroup was created to encapsulate respondents who teach in multiple clusters. There were participants who instructed in two and even all three grade clusters. The quantity of participants in each subsection, according to the order listed above, is the following: 55, 38, 39, and 27. Administrators were excluded from this set of analysis because they do not teach in the classroom and their subgroup was already analyzed in the first set of subdivisions.

Table 4

Descriptive Statistics on Understanding of Coding (Based on Level of Instruction)

Item	Elem			Mid			High			MC		
	M	*SD*	MD	*M*	*SD*	MD	*M*	*SD*	MD	*M*	*SD*	MD
Girls are generally less involved in coding compared to boys in school.	2.71	0.85	3	2.66	0.91	3	2.49	0.79	3	2.89	1.31	2
There are specific ethnic minorities that lack exposure to coding instruction in schools.	2.05	1.03	1	1.63	0.85	1	2.09	0.93	2	1.85	1.13	1
It is valuable to learn coding.	1.73	0.78	1	1.71	0.61	2	2.03	0.71	2	1.33	0.68	1
Male and female students can succeed in coding classes.	1.16	0.46	1	1.08	0.27	1	1.13	0.47	1	1.11	0.42	1
There is a gap in male and female Representation in STEM occupations.	2.24	1.09	3	2.00	0.93	2	1.90	0.97	1	1.67	0.78	1
There is a gap in male and female interest in STEM subjects in K–12 Catholic schools.	3.04	1.15	3	2.63	1.28	2	2.59	0.79	3	2.63	1.11	3
There is a gap in interest in K–12 STEM subjects between different ethnic groups.	2.65	1.19	3	2.34	1.15	2	2.46	0.91	3	2.33	1.24	1
There is a divide in STEM occupation representation between different ethnic groups.	2.22	1.01	3	1.89	0.89	1	1.90	0.91	1	1.93	0.87	2
I have a general understanding of what coding is.	2.29	0.96	2	1.97	1.03	2	1.95	1.05	2	1.78	0.97	1

Table 4 (continued)

Descriptive Statistics on Understanding of Coding (Based on Level of Instruction)

Item	Elem			Mid			High			MC		
	M	*SD*	MD	*M*	*SD*	MD	*M*	*SD*	MD	*M*	*SD*	MD
I have a more detailed understanding of what coding entails.	3.58	1.32	4	3.24	1.38	4	3.18	1.45	4	2.56	1.50	2

Note. MC = multiple cluster. MD = mode. For purposes of brevity, this portion of the last question in the table is included here: (examples: I understand how some coding apps work, I have used some coding apps myself, or I have taken coding courses in school).

The first item evaluated under the umbrella of the first research question dealt with whether there is less female involvement in coding relative to boys. The means for the subdivisions were 2.71, 2.66, 2.49, and 2.89. The modes for the first three subdivisions were *neither agree nor disagree* and it was *somewhat agree* for the last cluster. The standard deviations were 0.85, 0.91, 0.79, and 1.31. All the means, especially for the first three subgroups, trended toward the middle point of the scale and their modes demonstrated a relative ambivalence toward this question. The standard deviations below one also suggested a reasonable clustering of scores around the mean. The outlier in this group was the multiple cluster subdivision. Although their mode was *somewhat agree*, it also had the highest mean of 2.89 and standard deviation of 1.31. This indicates there were responses away from the mean. Furthermore, this subgroup's population must also be considered. It contained some technology and CS teachers and also had the least number of participants compared to the other three subgroups. The data may simply suggest some belief that female involvement is not lacking in coding. With the other subgroups' overall data points, there did not seem to be a strong sentiment one way or another. Responses showed uncertainty regarding this statement.

If a participant responded with *strongly agree* or *somewhat agree*, they were then prompted to respond to the reason why girls are less involved in coding. The response choices were *less interest* (1), *societal impact resulting from perceptions that coding is more suitable for males* (2), *both choices* (3), and an option for an alternate text response, notated by *other* (4). 20 participants from the elementary group responded to this question and the percentages of 20%, 35%, and 45% underscore the strong belief that less interest and societal factors impede female involvement in coding. No participant offered an alternative explanation. The middle school cluster had 16 responses. The resulting percentages were 12.50%, 31.25%, 43.75%, and 12.50%, which indicate this subgroup as well deemed less interest and societal factors as the predominant hinderances. Two respondents mentioned less access and historical discouragement of females in STEM areas. This latter response truly falls under the second response choice offered, which was societal impact because discouragement is directly connected to that reason. The high school subsection had 19 respondents with percentages of 10.52%, 52.63%, 26.31%, and 10.52%. This group believed societal factors were the greatest obstacles to female coding involvement. Moreover, two participants offered an alternative reason girls have had historically less exposure to coding. Finally, the last subgroup had 14 respondents for this question with frequencies equating to percentages of 7.14%, 35.71%, 50%, and 7.14%. Like the previous three subdivisions, societal factors and less interest, as indicated by the 50% mark for the third option, were perceived to the predominant choices for reduced female involvement with coding.

An additional item queried whether there is a gap between female and male interest in STEM subjects in K–12 Catholic schools. The means for these items were 3.04, 2.63, 2.59, and 2.63 and the modes, except for the middle school level of *somewhat agree*, were all *neither* *agree nor disagree*. The standard deviations were 1.15, 1.28, 0.79, and 1.11. These numbers essentially exhibit uncertainty, which is difficult to explain. Participants believed societal factors diminished female involvement in coding relative to boys; however, when asked about STEM interest, participants showed some unclarity. This may be due to the belief that female interest in STEM, which encompasses a much broader array of subjects or disciplines, is greater relative to coding. However, this may not be an adequate explanation either because responses hovered around the center of the scale. If participants believed there was no difference, their responses would trend toward the end of the scale. This may simply indicate there is some perceived gap between female and male STEM interest. The subsequent item continued to provoke thought on STEM, but specifically focused on whether there is a gap in occupational representation between genders. The means were 2.24 (mode of *strongly agree*), 2 (mode of *somewhat agree*), 1.90 (mode of *strongly agree*), and 1.67 (mode of *strongly agree*). The standard deviations were 1.09, 0.93, 0.97, 0.78. The standard deviations of around 1.0 demonstrated some responses were not in accord with the mean; however, taken as a whole, the data points signify the perception that there is an issue with female representation in STEM occupations.

Albeit the obstacles females have faced with exposure to and involvement in coding, respondents nevertheless asserted that females can succeed in coding classes. The mean for this item were 1.16, 1.08, 1.13, 1.11 and all the modes were *strongly agree*. The standard deviations of 0.46, 0.27, and 0.47, and 0.42 indicate a preponderance of responses were close to the means. Literature has lucidly explained that females have the potential to succeed; however, access and exposure are critical to enhance interest and involvement. These data points are in congruence with these sentiments.

Other than evaluating perceptions regarding female involvement in coding, the survey also assessed whether specific ethnic minorities have experienced difficulties with access to coding and STEM. The first question in this line of queries asked whether different ethnic groups have had less exposure to coding. The means were 2.05, 1.63, 2.08, and 1.85 for the respective subdivisions and the modes were all *strongly agree* except for the high school response of *somewhat agree*. The standard deviations were 1.03, 0.85, 0.93, and 1.13. The means hovering around 2.0 and the modes of *strongly* or *somewhat agree* signify fair agreement in certain ethnic minorities' lack of exposure to coding. The standard deviations around 1.0 demonstrate some divergence away from the mean, but the totality of the data points show reasonable agreement with the item's statement.

The survey also queried whether respondents believe there is a gap in interest in K–12 STEM subjects between different ethnic groups. The means were 2.65, 2.34, 2.46, and 2.33, with modes of *neither agree nor disagree*, *somewhat agree*, *neither agree nor disagree*, and *strongly agree*. The standard deviations were 1.19, 1.14, 0.91, and 1.24. The numbers across all three measures for this question are greater than the ones regarding exposure to coding, which may indicate students in different ethnic groups are interested in STEM subjects, but they lack the exposure in their schools. However, not all the statistics were consistent. For instance, the plurality of teachers in the multiple grade clusters subgroup chose *strongly agree* with the statement, yet the mean was relatively the same as others. The standard deviation was the highest relative to the other groups, which means that although more respondents chose *strongly agree*, there were also a tangible number who chose responses away from the mean.

The final question in this series gauged perceptions regarding a divide in STEM occupation representation between different ethnic groups. The respective means were 2.22, 1.89, 1.90, and 1.93. The elementary subgroup's mode was *neither agree nor disagree*, which caused the mean to be slightly higher than the other groups. The next two groups chiefly chose *strongly agree* and the multiple cluster subdivision's mode was *somewhat agree*. The standard deviations were 1.01, 0.89, 0.91, and 0.87. The groups' general sentiment was that there is a disparity in STEM representation between different ethnic groups.

All four subdivisions were also asked about their general and more nuanced understanding regarding coding. With respect to general understanding, the first three subgroups had means close to 2 and a mode of *somewhat agree*; however, the multiple cluster subgroups had a lower mean of 1.78 and a mode of *strongly agree*. Once again, this may be due to this group membership, which is partly made of technology and CS teachers. All four standard deviations were either 1.0 or very close to it. The indicators for perceptions about detailed understanding of coding were higher across all four subdivisions. The means were 3.58, 3.24, 3.18, and 2.56, and the modes were all *somewhat disagree* except for the final group's indication of *somewhat agree*. The standard deviations were 1.32, 1.38, 1.45, and 1.50. The first three subgroups' indicators suggest, although more people perceived themselves uninformed or unacquainted about the nuances of coding, there were some who believed they comprehended its intricacies as well. The last subgroup was generally more confident in their knowledge of coding, but it also had the highest standard deviation. This indicates some scores were down the scale far away from the mean. In totality, the statistics resulting from both questions show a general need to develop the participant understanding of coding, which is paramount to properly instructing it in the classroom. To be sure, simply enhancing their knowledge is insufficient. Pedagogical tools also must be developed to facilitate its instruction properly; however, enhancing participants' understanding of what coding is serves as a crucial foundation.

Analysis of Teachers in Different School Levels and Administrators. The final set of subgroups analyzed for the first research question, which were composed of educators at the elementary, middle, and high school levels, along with those who instruct in multiple bands, and administrators, were analyzed descriptively and inferentially based on a proposed set of five constructs. The isolated number of participants in each respective subgroup for all constructs can be referred to in Table 5. The constructs were created from survey items that were most related to each research question.

Table 5

Subgroup Sample Size

Subgroup	*n*	%
Elementary	54	29
Middle	35	18.8
High	34	18.3
Multiple cluster	25	13.4
Administrator	38	20.4
Total	186	100

There was a discrepancy in participant responses, which posed challenges in organizing the data. Question 65, which asked about the participant's role or position in school, yielded 38 administrators. However, Question 66, which asked what grade level the participant instructs, generated 27 administrators. Response choices included one allowing participants to choose the administrator option to indicate they do not teach in the classroom; however, there were 11 individuals who provided contradictory responses. Eleven individuals stated they were administrators in Question 65 but indicated they instructed in Question 66. This was confusing because there was a clear option to select for administrators. Although participants were allowed to choose multiple responses, none of those 11 individuals stated they were administrators in Question 66. I originally compiled the data for this set of analysis with 197 participants; however, I reached that *n* because I used the administrator responses from Question 65 (38) and the grade level of instruction's responses from Question 66. Consequently, I had 38 administrators in the data, but I also used those 11 individuals' responses from Question 66 stating they were teachers. Thus, I counted 11 people twice. Once I discovered this issue, I identified those 11 individuals, removed them from the data set, and reconducted the descriptive tests.

For this set of analysis, I decided to use the administrator responses from Question 65 because that was the first question asking about a participant's role. I was miffed at the discrepancy between the two questions, but I trusted Question 65's responses more because it came before Question 66, and the nature of the question provided a clear delineation between roles and the ability to choose multiple responses. Thus, I assumed those 38 responses were truly from administrators. For the remainder of the data, I used information from Question 66 because it was more in line with the subgroups I focused on in this set of analysis (elementary, middle, and high school, and participants teaching in multiple grade clusters). Those who chose multiple grade bands were placed in the multiple cluster subgroup. Five people did not respond to either Question 65 or 66 and thus were excluded from this specific analysis. The composition of each group is noted in Table 5.

Analysis of First Construct. The first research question's construct consisted of five items (see Table 6), including whether participants have a general understanding of coding, a more detailed understanding of what it entails such as how some coding applications function and if they have taken coding courses in school, whether they can effectively teach block-based coding programs like Scratch and syntax-based coding programs like Python, and to what degree they are adept at integrating coding into the content curriculum. Cronbach's alpha for the first construct was .84, which surpassed the .70 threshold for reliability. This indicated the items were interrelated and thus could sufficiently explain participants' understanding of coding. Furthermore, response choices ranged from 1 to 5 (see Table 6). Furthermore, all items met the item-total correlation threshold of 0.2, which is indicative of the degree of interrelationship between the inquiries (P. Ruengvirayudh, personal communication, March 14, 2022). This number ranges from -1 to 1 (P. Ruengvirayudh, March 14, 2022).

Table 6

Construct 1—Understanding of Coding

Survey Item	Response Option	α
		.84
I have a general understanding of what coding is.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
I have a more detailed understanding of what coding entails.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
How effectively can you teach block-based coding programs like Scratch?	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
How effectively can you teach syntax-based coding programs such as Python?	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
I can integrate coding into my content curriculum.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	

Prior to calculating Cronbach's alpha, I recoded questions 48 and 49, which are the third and fourth questions in the table. They had a different scale relative to the other items in the construct; however, it is crucial that all items in a construct are assessed with an identical scale

(P. Ruengvirayudh, March 10, 2022), otherwise the data will not display accurate perceptions of participants. The original scale had four levels of measurement, including *highly capable* (1), *capable* (2), *somewhat capable* (3), and *not capable* (4). The scale for the other items in the construct, which was the same for almost every item in the survey, were *strongly agree* (1), *somewhat agree* (2), *neither agree nor disagree* (3), *somewhat disagree* (4), and *strongly disagree* (5). The responses were recoded to the following: 1 = 1, 2 = 2, 3 = 4, and 4 = 5. *Highly capable* was equivalent to *strongly agree*, *capable* can be regarded as *somewhat agree*, there was no neutral position in the 4-point scale, *somewhat capable* was assessed as *somewhat disagree*, and *not capable* was commensurate with *strongly disagree*. *Somewhat capable* and *somewhat disagree* were likened because the former option can indicate the person is somewhat not capable as well. Thus, it would be proportionate to *somewhat disagree*.

The means and standard deviations for the five items are in Table 7. The question gauging participants' general understanding of coding received the lowest mean; however, their more nuanced understanding of coding waned in comparison. Furthermore, questions assessing their pedagogical competence, or ability to instruct coding in the classroom, received the highest means demonstrating their perceived inability to instruct block- or syntax-based coding in the classroom. Standard deviations for every question rose above 1.0; therefore, there were responses that digressed away from the mean, including those that perceived themselves as capable of instructing coding. Nevertheless, the overall picture of the data signified participants' general lack of understanding of coding.

Table 7

Descriptive Statistics for Construct 1

Survey Item	*M*	*SD*	*n*
I have a general understanding of what coding is.	2.08	1.02	186
I have a more detailed understanding of what coding entails	3.3	1.42	186
How effectively can you teach block-based coding programs like Scratch?	4.05	1.46	186
How effectively can you teach syntax-based coding programs such as Python?	4.44	1.15	186
I can integrate coding into my content curriculum.	3.54	1.42	186

The entire construct's evaluation according to the five subgroups, as emphasized in Table 8, demonstrated the multiple cluster group deemed themselves most understanding and capable of instructing coding in the classroom. The simple, yet important reason for this outcome is that this group was partly comprised of teachers who teach CS or technology. Among those who indicated the subjects they teach, four individuals in this groups specified those courses. It is also entirely possible others in this subdivision can do the same. The other four subgroups were in approximately the same area of the scale's spectrum and their mean for this construct was 3.58, suggesting a diminished degree of understanding and a relative discomfort with instructing coding. The average mean of the five groups for this construct was 3.48, which was slightly lower relative to the mean of 3.58, due to the multiple cluster group.

Table 8

Descriptive Statistics for Construct 1's Subgroups

Subgroup	*M*	*SD*	*n*
Elementary	3.73	0.87	54
Middle	3.41	0.89	35
High	3.5	1.05	34
Multiple cluster	2.69	1.24	25
Administrator	3.69	0.88	38
Total	3.48	1.01	186

Inferential statistics were also conducted to better understand whether there is a difference in means between the five subgroups. Data can be viewed in Table 9. Assumptions such as normality and homogeneity of variance were assessed and because they were violated, it is imperative to read the data with caution (P. Ruengvirayudh, personal communication, March 13, 2022). The data were based on 186 participants' responses with no missing values and were organized into five groups. The elementary subgroup had 54 participants (29.03%), the middle school subdivision had 35 (18.81%), the high school participants were 34 in number (18.27%), the multiple cluster subsection had 25 respondents (13.44%), and there were 38 administrators (20.43%). Response choices for Construct 1, which evaluated participants' understanding of coding, ranged from 1 to 5 (M = 3.48, SD = 1.01). Table 6 indicates the Welch test's p value of .005, which resulted in rejecting the null hypothesis. It thus indicated there is at least one statistically significant difference in means between groups: Elementary (M = 3.73, SD = 0.87), Middle (M = 3.41, SD = 0.89), High (M = 3.5, SD = 1.05), Multiple Cluster (M = 2.69, SD = 1.24), and Administrator (M = 3.69, SD = 0.88), Welch's F (4, 79.83) = 4.0, p =.005, η_2 =.11.

Table 9

Robust Tests of Equality of Means for Construct 1

	Statistic	*df1*	*df2*	*p*	η_2
Welch	4.00	4	79.83	.005	0.11

Note. df = degrees of freedom.

Post-hoc tests using Games-Howell, which is identified in Table 10, demonstrated the elementary group was significantly different than the multiple cluster group (p = .005), and the multiple cluster subdivision was significantly different than the group of administrators (p = .009). The only other point-wise comparison that came remotely close to the .05 threshold was the high school and multiple cluster group (p =.076). All other comparisons were significantly above the .05 p-value threshold, which indicates a lack of a statistically significant difference between the means of the groups. According to P. Ruengvirayudh (personal communication, March 13, 2022), based on Cohen's guideline to interpret the effect size (eta squared or η_2), the point estimate value of.11 was between moderate and large ($r2$ =.01 is small; $r2$ =.09 is medium; $r2$ = .25 is large). This indicates the subgroups impacted respondents' understanding of coding. Ultimately, these data suggest, other than the statistically significant differences in means noted above, all other groups' understanding of coding were relatively the same. To be sure, this does not indicate their levels of understanding are identical; rather, it signifies a lack of a statistically significant difference.

Table 10

Post Hoc Results for Construct 1 (Games-Howell)

Group	*M*	*SD*	Comparison group	Mean difference	*p*
Elementary	3.73	0.87	Middle	0.32	0.450
			High	0.23	0.814
			Multiple Cluster	1.05	0.005
			Administrators	0.04	0.999
Middle	3.41	0.89	Elementary	-0.32	0.450
			High	-0.09	0.996
			Multiple Cluster	0.72	0.110
			Administrator	-0.28	0.664
High	3.5	1.05	Elementary	-0.23	0.814
			Middle	0.09	0.996
			Multiple Cluster	0.81	0.076
			Administrator	-0.19	0.922
Multiple Cluster	2.69	1.24	Elementary	-1.05	0.005
			Middle	-0.72	0.110
			High	-0.81	0.076
			Administrator	-1.00	0.009
Administrator	3.69	0.88	Elementary	-0.04	0.999
			Middle	0.28	0.664
			High	0.19	0.922
			Multiple Cluster	1.00	0.009
Total	3.4	1.01			

Note. Sig.= significance.

First Research Question Conclusion

The data from the survey's entire population illustrates a firm belief that coding is valuable and that students would be motivated to learn it. However, educators believed access to technology is inequitable in Catholic schools and that ethnic minorities are adversely impacted as suggested by their disproportionate exposure to coding instruction. This perception was in harmony with the belief that there is unequal interest in STEM subjects among certain ethnic groups and that diversity in such occupations is lacking.

Regarding the differences between males and females, respondents predominantly believed females were less represented in STEM occupational fields, but interest among females was not as lacking relative to different ethnic groups. Respondents also affirmed both genders can succeed in coding and there was no discernible difference in one gender's greater aptitude relative to the other. As such, the perception surrounding female students and coding was more encouraging relative to ethnic minorities. Furthermore, respondents showed some agreement with the notion that there is a disparity in interest in K–12 STEM subjects and representation in such occupations between different ethnic groups.

The data on the teacher, administrator, and multiple role subgroups also produced noteworthy results. Generally, the multiple roles subgroup trended closer to strongly agreeing with the items, whereas the teacher and administrator subgroups were more closely aligned. Nevertheless, there was a relatively firm perception of coding's value to students and the potential for female and male students to succeed. Furthermore, respondents did not strongly believe there was a gap in interest in STEM subjects and representation in STEM occupations between males and females and different ethnic groups. When the subgroups were prompted to perceive their understanding of coding, all three showed a relatively strong understanding of what coding generally entails; however, when prompted to opine on their more nuanced understanding of coding, the confidence levels decreased. However, the last group's perception remained relatively steady, which indicates they held the strongest understanding of coding among the three groups.

Data on the female and male subgroups showed females were more likely to believe there is less involvement among females in coding and they cited societal factors as the chief impediment. Males did show some agreement with the statement; however, their mean trended toward the middle of the scale. Furthermore, both groups considered coding to be valuable and that females can succeed in this area of CS. Regarding a potential gap in interest between male and female STEM interest, both groups did demonstrate some agreement with the statement, but their overall responses more so veered toward the middle of the scale, which may indicate indecisiveness on the issue. Furthermore, both groups, especially females, asserted there is a gap in representation in STEM occupations. As for questions regarding ethnic minorities, both groups were asked if there is a gap in interest in K–12 STEM subjects between different ethnic groups. Both groups did affirm a divide exists, but the males were less resolute in the strength of their responses. Regarding representation in occupations, both subgroups' data suggested their perception that there is a disparity. Finally, when prompted to opine on their general and then more nuanced understanding of coding, males perceived themselves as more comfortable on both items, but both subgroups' confidence levels decreased with the second question.

The isolated data based on clustered grade levels demonstrated a relative ambivalence regarding female involvement in coding relative to boys. Although this was unexpected, the reasoning for it must be further investigated. However, those who did recognize an imparity between male and female involvement generally cited societal factors and less interest as the causes. Similarly to the uncertainty concerning female involvement compared to males in coding, the general perception regarding female interest in K–12 STEM subjects was also indecisive. There was no strong sentiment either way. However, this changed when participants were queried on female representation in STEM occupations and ability to succeed in coding classes. Participants across all four subdivisions responded with agreement toward a lack of female representation in STEM occupations and an ability for females to comparably succeed in coding.

Similar lines of questioning regarding ethnic groups were posed. For instance, there was a modicum of agreement toward the existence of a divide in interest between different ethnic groups in K–12 STEM subjects, but it was not pronounced. However, regarding STEM occupation representation, quite similarly to the perceived lack of diversity between males and females, respondents believed this to be the case between different ethnic groups.

To further this study's analysis of participants' understanding of coding, they were candidly asked about their general and nuanced levels of comprehension. Participants in all subgroups had a better understanding of what coding generally entails; however, when prompted to assess their degree of a more detailed understanding, confidence levels in al subdivisions diminished.

The analysis of the final set of five subgroups according to the first construct, which had a Cronbach's alpha of .84, indicated the multiple cluster group possessed the highest level of understanding and capacity to instruct coding, relative to the other four groups. Interestingly, there was no distinguishable difference between the means of the other four groups as they each hovered around a mean of 3.5. This demonstrates an equitably meager understanding of coding ability and ability to instruct it. The inferential statistics corroborated these overarching points to the extent that the differences in means between the multiple cluster and elementary group were statistically significant ($p = .005$). Furthermore, another statistically significant difference in means was noted between the multiple cluster and administrator subgroups ($p = .009$). This difference was slightly less relative to the difference between the multiple cluster and elementary group, but sufficient, nonetheless. The difference in means between the multiple cluster and high school group was also noteworthy with a p value of .08, but it did not cross the threshold for significance. Nevertheless, this piece of data shows some difference between these two groups. Ultimately, the data demonstrate the multiple cluster group's understanding of coding is superior.

Second Research Question

Once participants' level of understanding was assessed, I focused on whether coding instruction provides tangible benefits for students and if it can diversify the STEM fields. Consequently, I posed the following question: What is their perception on coding instruction's impact on students' academic achievement, occupational preparedness, and on its ability to enhance STEM representation in school and in the workforce?

Survey Results

The following section will provide a robust discussion explaining the results of the various analyses conducted as part of the second research question, including, but not limited to, an assessment of the entire set of data, an analysis of data based on the position of the respondents, and an evaluation of responses based on the participants' grade level of instruction.

Analysis of Total Sample Population. The second research question analyzed three different perceived benefits resulting from coding instruction. Table 11 identifies the mean, standard deviation, and mode for the questions corresponding to the perceptions of benefits as it related to academic achievement. These questions, which inquired into views on coding's impact on abilities such as logical thinking and creativity, were composed based on the prevailing literature that discussed these themes.

Table 11

Total Sample Population's Descriptive Statistics for Second Research Question's Items (Academic Achievement)

Survey Item	*M*	*SD*	Mode
Coding develops a student's logical thinking ability.	1.45	0.69	1
Coding develops a student's problem-solving ability.	1.42	0.63	1
Coding develops a student's creativity.	1.53	0.75	1
Coding develops a student's collaboration skills.	1.85	0.81	2
Coding develops a student's social skills.	2.33	0.96	2
Learning coding can help a student perform better in different subjects.	1.85	0.84	1
Learning coding helps a student be more engaged in school.	2.01	0.84	2
Coding is an important 21st century literacy.	1.62	0.81	1

Respondents resoundingly deemed coding to enhance a student's logical thinking and problem-solving abilities as both statements had a mean of slightly above 1.4, a standard deviation between 0.6 and 0.7, and a mode of *strongly agree* (1). The standard deviation indicates responses hovered around the mean, yet there were some that strayed somewhat away. Additionally, participants asserted coding cultivates a student's creativity. The measures were very close to those stemming from the first two statements in the table. The mean of 1.53, a standard deviation of 0.75, and a mode of *strongly agree* (1) suggest a close linkage between coding and development of creativity. Respondents were less bullish on coding's influence on a student's collaboration and social skills. The mean and standard deviation for the former were 1.85 and 0.81, and 2.33 and 0.96 for the latter query. The mode for both statements was *somewhat agree* (2). Although they considered coding as a constructive influence on these abilities, the respondents did not recognize it as viable relative to its impact on logical thinking and problem-solving. Respondents also believed coding facilitates a student's progression in different subjects as evidenced by a mean of 1.85, a standard deviation of 0.84, and a mode of *strongly agree* (1). However, when asked if coding assists in student engagement, the mean rose to slightly above 2.0 with a standard deviation of 0.84. Although the means were very close and the standard deviations essentially identical, the mode for the student engagement statement was *somewhat agree* (2). It is interesting to assess why more respondents strongly believed coding improved a student's capacity to perform well in different subjects, but they *somewhat agreed* with coding's ability to augment student engagement. According to the extant literature, student engagement is inculcated because they consistently collaborate during the coding process. Although there is not a great discrepancy in the central measures with these queries, it is still noteworthy. Respondents may have responded to this question with general school engagement in mind, yet that still seems a bit inconsistent because they strongly believed coding facilitates improved student achievement in different courses. The final question in this block asked participants to respond to whether coding is an important literacy. The mode was *strongly agree* (1), the mean was 1.62, and the standard deviation was 0.81. Because this was a general statement, the answers validate the assessment that coding is crucial for students to learn.

The second research question additionally evaluated the respondents' insights regarding coding's capacity to prepare students for different occupations in the future. Table 12 underscores the measures of central tendency for this component of the question.

Table 12

Total Sample Population's Descriptive Statistics for Second Research Question's Items (Occupational Preparedness)

Survey Item	*M*	*SD*	Mode
Learning coding in K–12 will better prepare students to earn positions in technology specific fields.	1.43	0.63	1
Learning coding in K–12 will better prepare students to earn positions in non-technology specific fields, such as medicine and retail.	2.04	0.91	1

Although two questions were focused upon, it nevertheless yielded significant data. Comparing both questions, respondents believed coding will better prepare students for technology related positions relative to non-technology-specific occupations. The mean for the former statement was a stout 1.43 with a standard deviation of 0.63 and a mode of *strongly agree* (1), whereas the latter statement generated a more tepid confidence in coding's ability to prepare students for positions in fields such as government and medicine. Although the mode was still *strongly agree* (1) for this query, the mean was slightly over 2 with a larger standard deviation reaching almost 1. This does not indicate respondents do not believe in coding's ability to prepare students for non-technology-related positions, but there is a stronger linkage in perceptions between coding and occupational preparedness in the technology sector. However, because the literature indicates traditional occupational sectors are incorporating more technology, immersing students in coding will better position them to also be employed in such fields.

The final prong of the second research question homed in on the question of social justice. The survey evaluated respondents' positions on coding's potential to enhance an interest in STEM within females and different ethnic groups and whether it can diversify representation in such occupational fields. Table 13 highlights the measures used to gauge the data for this part of the question.

Table 13

Total Sample Population's Descriptive Statistics for Second Research Question's Items (STEM Representation)

Survey Item	*M*	*SD*	Mode
Coding can help bridge the gap in K–12 STEM interest between males and females.	1.8	0.85	1
Coding can help bridge the gap in K–12 STEM interest between different ethnic groups.	1.94	0.93	1
Exposure to coding in K–12 schooling will help bridge the gap in STEM occupations held by males and females.	1.72	0.78	1
Exposure to coding in K–12 schooling will help bridge the gap in representation amongst different ethnic groups in STEM occupations.	1.8	0.87	1

Participants opined that they agreed in coding's potential to spur interest in STEM subjects. Educators believe the gap in STEM interest between males and females will diminish due to coding instruction as indicated by a mean of 1.80, a standard deviation 0.85, and a mode of *strongly agree* (1). Comparably, respondents believed in coding's capacity to bridge the gap in STEM interest between different ethnic groups. The mean for this statement was 1.94, the standard deviation was 0.93, and the mode was *strongly agree* (1). Although the mean and standard deviation are slightly higher relative to the same question regarding females and males, it is nonetheless a substantial marker. The literature suggested minority populations have disproportionate access to technology and coding instruction; therefore, offering it to all can potentially reduce the current disparity that exists and thus serve as a tangible step toward greater social justice.

The two statements connecting coding instruction to diversifying representation in STEM occupations was even more robust relative to the first two statements. Participants' responses yielded a mean of 1.72, a standard deviation of 0.78, and a mode of *strongly agree* (1) suggesting coding can improve representation of females in STEM occupations. Similarly, participants believed in coding instruction's potential to mitigate the imparity in STEM representation between different ethnic groups as indicated by a mean of 1.8, a standard deviation of 0.87, and a mode of *strongly agree* (1). Although the means of 1.72 and 1.8 are quite close, the difference may be attributed to other considerations beyond the availability of coding instruction. For instance, the respondents may believe this instruction may need to be combined with other measures to create substantive change in STEM occupation representation among different ethnic populations.

Analysis of Subgroups Based on Position. Several items that corresponded to the second research question were also analyzed according to the three subgroups of teachers, administrators, and personnel with multiple roles. The data are located in Table 14.

Table 14

Descriptive Statistics on Benefits of Coding (Based on Participants' Position)

Survey Item	Teacher			Admin			MR		
	M	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode
Coding develops a student's logical thinking ability.	1.48	0.69	1	1.42	0.60	1	1.34	0.67	1
Coding develops a student's problem-solving ability.	1.48	0.68	1	1.39	0.55	1	1.18	0.39	1
Coding develops a student's creativity.	1.64	0.83	1	1.5	0.56	1	1.18	0.46	1
Coding develops a student's collaboration skills.	1.99	0.83	2	1.76	0.79	1	1.55	0.65	1
Learning coding can help a student perform better in different subjects.	1.82	0.86	1	1.82	0.80	2	1.71	0.84	1
Learning coding in K–12 will better prepare students to earn positions in technology-specific fields.	1.47	0.65	1	1.55	0.69	1	1.26	0.50	1
Learning coding in K–12 will better prepare students to earn positions in non-technology specific fields, such as medicine, retail, and government.	2.03	0.90	1	2.08	0.91	2	1.95	0.90	2
Exposure to coding in K–12 schooling will help bridge the gap in STEM occupations held by males and females.	1.79	0.84	1	1.76	0.68	2	1.5	0.65	1
Exposure to coding in K–12 schooling will help bridge the gap in representation amongst different ethnic groups in STEM occupations.	1.88	0.95	1	1.95	0.77	2	1.42	0.60	1

Teachers quite profoundly believed coding develops students' logical thinking ability. The data produced a mean of 1.48, a standard deviation of 0.69, and a mode of *strongly agree*.

The administrators' perceptions were even more effusive of coding's ability to develop logical thinking as evinced by a mean of 1.42, a mode of *strongly agree*, and a standard deviation of 0.60. The multiple role group was the most inclined to agree with this statement because the mean was 1.34, the mode was *strongly agree*, and the standard deviation was .67, which does suggest some deviation away from the mean. The mean and standard deviation for the teacher subgroup as it relates to coding's ability to develop a student's problem-solving capacity was essentially identical to its potential to inculcate logical thinking. The administrator subgroup's data were very similar to that of the teacher regarding this item and relative to the previously referred item regarding logical thinking, the administrators perceived coding to be even more effective at facilitating the development of problem-solving skills. However, the subgroup of personnel with multiple roles was the most ebullient regarding coding's capability to improve problem-solving skills as indicated by a mean of 1.18, a standard deviation of 0.39, and a mode of *strongly agree*. As for fostering student creativity, all three subgroups provided parallel responses that inclined toward the *strongly agree* response choice. The item that did not receive nearly as much support regarded coding's penchant to form collaboration skills. Teachers' responses returned a mean of 1.99, a standard deviation of 0.83, and a mode of *somewhat agree*. This does not suggest teachers do not believe collaboration skills can be fostered; however, responses to development of other skills were significantly more effusive. The administrators, however, were more enthusiastic about coding's capacity to foster collaboration skills. The mean for their responses was 1.76, the mode was *strongly agree*, and the standard deviation was 0.79, which demonstrates some variance away from the mean. The responses from participants with multiple roles were once again most fervent as indicated by a mean of 1.55, a mode of *strongly* *agree*, and a standard deviation of 0.65. This can be attributed, at least partly, to the technology and CS teachers who were in this subgroup.

In addition to the prospect of developing diverse skills, participants were also surveyed on their perceptions regarding coding's capacity to help student achievement in different courses, better prepare students for various technology and non-technology specific occupations, and to bolster female and different ethnic groups' representation in STEM occupations. Regarding coding's potential to improve student achievement in different subjects, teachers suggested it does as indicated by a mean of 1.82, a mode of *strongly agree*, and a standard deviation of 0.86. The administrators' responses generated an identical mean of 1.82, but the mode was *somewhat agree*. The difference in modes between the teachers and administrators but an identical mean for the latter suggests more administrators chose *somewhat agree*, but there was a bit more variation with the teachers' responses. The multiple role subgroup was relatively the most inclined to believe in coding's potential to assist students in different subject. The mean of 1.71, mode of *strongly agree*, and standard deviation of 0.84 is evidence of this contention.

When prompted to respond to whether coding instruction will help students obtain technology and nontechnology specific positions, all three subgroups believed the former is a greater possibility. The teachers' mean for technology positions was 1.47, but 2.03 for nontechnology positions. The administrators' means were 1.55 and 2.08, respectively, while the multiple subgroups' means were 1.26 and 1.95. Furthermore, the standard deviations for all three groups increased from the technology to non-technology-related positions questions, while the modes remained consistent between *strongly agree* and *somewhat agree*. These data by no means suggest these subgroups do not believe coding will help improve the prospects for obtaining non-technology-related position; however, they perceive coding to better assist students in procuring technology-centered positions.

The last set of questions refocused attention upon gender and ethnic backgrounds. Specifically, participants were queried on whether exposure to coding instruction in school will diversify female and ethnic representation in STEM occupations. Teachers' replies netted a mean of 1.79, a mode of *strongly agree*, and a standard deviation of 0.84. Administrators' responses generated a mean of 1.76, a mode of *somewhat agree*, and a standard deviation of 0.68. Finally, the multiple roles subgroup produced a mean of 1.5, a mode of *strongly agree*, and a standard deviation of 0.65. These data suggest all three groups believe coding can help diversify female representation in STEM occupation. Furthermore, these data continue the pattern of the multiple roles subgroup holding smaller means and standard deviations.

Teachers also somewhat believed in coding's ability to diversify the representation of ethnic groups in STEM occupations as evidenced by a mean of 1.88, a mode of *strongly agree*, and a standard deviation of 0.95. Administrators' responses were somewhat comparable to the teachers' data as suggested by a mean of 1.94 and a mode of *somewhat agree*. The final subgroup more closely aligned with the *strongly agree* response choice as indicated by a mean of 1.42, a mode of 1, and a standard deviation of 0.60.

Analysis of Subgroups Based on Gender. The study gauged participants' perceptions on the benefits of coding instruction based on gendered subgroups (see Table 15). There were 141 female participants, 43 males, five did not respond to the item asking to identify gender, and two selected the decline to state option.

Table 15

Descriptive Statistics on Benefits of Coding (Based on Gender)

Survey Item	Female			Male		
	M	*SD*	Mode	*M*	*SD*	Mode
Coding develops a student's logical thinking ability.	1.39	0.62	1	1.51	0.70	1
Coding develops a student's problem solving ability.	1.35	0.55	1	1.51	0.67	1
Coding develops a student's creativity.	1.48	0.68	1	1.56	0.70	1
Coding develops a student's collaboration skills.	1.84	0.81	2	1.84	0.78	2
Learning coding can help a student perform better in different subjects.	1.84	0.85	1	1.84	0.78	1
Learning coding in K–12 will better prepare students to earn positions in technology-specific fields.	1.37	0.58	1	1.65	0.75	1
Learning coding in K–12 will better prepare students to earn positions in non-technology specific fields, such as medicine, retail, and government.	1.96	0.90	1	2.26	0.88	2
Exposure to coding in K–12 schooling will help bridge the gap in STEM occupations held by males and females.	1.60	0.73	1	2.12	0.82	3
Exposure to coding in K–12 schooling will help bridge the gap in representation amongst different ethnic groups in STEM occupations.	1.70	0.86	1	2.14	0.83	3

The first series of questions analyzed according to these two subgroups gauged whether coding instruction inculcates the general skills of logical thinking, problem solving, creativity, and collaboration within students. Keeping the listed order of these skills in mind, data from female responses indicated strong agreement with all except collaboration. Means for the first three skills were 1.39, 1.35, and 1.48, the mode was *strongly agree*, and the standard deviations were 0.62, 0.55, and 0.68. Latter pieces of data demonstrate a relatively strong gathering of scores around the mean. Regarding coding's advancement of collaboration, the mean of 1.84, mode of *strongly agree*, and standard deviation of 0.84 demonstrate considerable agreement, but the indices were not as potent as the ones for logical thinking, problem solving, and creativity. Data from male responses paralleled females' numbers with means of 1.51, 1.51, 1.56, a mode of *strongly agree*, and standard deviations of 0.70, 0.67, and 0.70. These data indicate a robust belief in coding's potential for developing these skills. Like females, however, the mean for coding's capacity to advance collaborative skills was not as high. To be sure, the mean of 1.84, the mode of *somewhat agree*, and standard deviation of 0.78 highlight sizable agreement; however, it was not as high as the other indices. The reasoning for this difference would need to be investigated further to provide an explanation, especially because coding typifies collaborative work. This may indicate participants do not necessarily realize collaboration can be developed due to coding.

The subsequent set of questions analyzed for these subgroups centered upon benefits as they relate to academic achievement, occupational preparedness, and diversification of STEM representation. As such, an item asked whether coding instruction enhances student performance in different subjects. Female participants' data generated a mean of 1.84, a mode of *strongly agree*, and a standard deviation 0.85, while the male participants' data produced a mean of 1.84, a mode of *strongly agree*, and a standard deviation of 0.78. Aside from the difference in modes, the figures are quite similar. Regardless of the slight difference, both groups generally believe coding instruction assists student achievement in the classroom.

The item that queried whether coding instruction will better prepare students to earn positions in technology-specific fields received firm support from both females and males. The female's data returned a mean of 1.37, a mode of *strongly agree*, and a standard deviation of 0.58, while the male participants' responses produced a mean of 1.65, a mode of *strongly agree*, and a standard deviation of 0.75. The male respondents' mean and standard deviation were slightly higher, which indicates there were some more responses that did not converge to the mean. Nevertheless, the overarching point of both subgroups' data projects a firm agreement in coding's capacity to equip students with the skills and background that are requisite to secure a technology-specific positions.

In regard to the question asking whether coding instruction better positions students to obtain non-technology specific positions, the indices were higher. Data from females yielded a mean of 1.96, a mode of *strongly agree*, and a standard deviation of 0.90, whereas male respondents' data produced a mean of 2.26, a mode of *somewhat agree*, and a standard deviation of 0.88. Although these results demonstrate a link between coding instruction and the ability to secure non-technology positions, the results indicate participants believe it better prepares students to obtain technology related positions. The literature, however, explicates traditionally non-technology-specific sectors such as government, retail, and medicine are shifting toward integrating various technologies and those that are adept at coding will be primed to earn positions in such fields. Participants, however, may not be aware of these shifting realities.

The final set of questions evaluated with this set of subgroups queried whether coding instruction can bridge the divide in STEM occupational representation between males and females and among different ethnic groups. The females' data produced means of 1.60 and 1.70, respectively, and a mode of *strongly agree*. The standard deviations were 0.73 and 0.86. The standard deviations suggest some responses were not as affirmatory; however, the mode and robust means indicate the female perception that coding instruction can enhance the representation of females and different ethnic groups in STEM occupations. However, the data from male responses was not nearly as affirmatory. The means for those two items were 2.12 and 2.14 with a mode of *neither agree nor disagree*. The standard deviations were 0.82 and 0.83.

These data points do not suggest males do not believe coding can diminish the imparity in representation, but they do not necessarily suggest they can. To be sure, the means trend close to *somewhat agree*, but the modes are neutral and standard deviations signify some responses deviate away from the mean. Additional investigatory analysis is required to determine the rational for the differences between female and male responses for these two questions.

Analysis of Subgroups Based on Different Grade Clusters. The study measured the perceptions of elementary, middle, and high school teachers as well as teachers in multiple grade-level clusters regarding coding's benefits. The descriptive data are presented in Table 16.

Table 16

Descriptive Statistics on Benefits of Coding (Based on Level of Instruction)

Item	Elem			Mid			High			MC		
	M	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode
Coding develops a student's logical thinking ability.	1.45	0.66	1	1.42	0.76	1	1.59	0.75	1	1.22	0.42	1
Coding develops a student's problem solving ability.	1.45	0.60	1	1.34	0.67	1	1.51	0.72	1	1.26	0.45	1
Coding develops a student's creativity.	1.55	0.72	1	1.42	0.83	1	1.69	0.86	1	1.37	0.56	1
Coding develops a student's collaboration skills.	1.89	0.74	2	1.79	0.96	1	2.10	0.79	2	1.67	0.78	1
Learning coding can help a student perform better in different subjects.	1.85	0.85	1	1.87	1.04	1	2.08	0.70	2	1.67	0.83	1
Learning coding in K–12 will better prepare students to earn positions in technology-specific fields.	1.42	0.63	1	1.42	0.60	1	1.62	0.75	1	1.15	0.36	1
Learning coding in K–12 will better prepare students to earn positions in non-technology specific fields, such as medicine, retail, and government.	1.89	0.90	1	2.18	1.01	1	2.38	0.67	2	1.67	0.92	1
Exposure to coding in K–12 schooling will help bridge the gap in STEM occupations held by males and females.	1.80	0.91	1	1.58	0.64	1	1.95	0.76	2	1.52	0.75	1

Table 16 (continued)

Descriptive Statistics on Benefits of Coding (Based on Level of Instruction)

Item	Elem			Mid			High			MC		
	M	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode
Exposure to coding in K–12 schooling will help bridge the gap in representation amongst different ethnic groups in STEM occupations.	1.82	1.02	1	1.71	0.80	1	2.00	0.79	2	1.63	0.84	1

Note. MC = multiple cluster. Ele = elementary

The first set of benefits related to general skills including logical thinking, problem-solving, creativity, and collaboration. Participants affirmed the advancement of logical thinking as a result of coding with means of 1.45, 1.42, 1.59, and 1.22 with a mode of *strongly agree*. The standard deviations of 0.66, 0.76, 0.75, and 0.42 showed a relatively small divergence of scores away from the mean. Observations regarding problem-solving ability were similarly stout with means of 1.43, 1.34, 1.51, and 1.26 in addition to a mode of *strongly agree*. The standard deviations of 0.60, 0.67, 0.72, and 0.45 exhibited some convergence around the mean. Responses for coding's propensity to augment creativity within students earned means of 1.55, 1.42, 1.69, and 1.37 alongside a mode of *strongly agree*. The standard deviations of 0.72, 0.83, 0.86, 0.56, however, were slightly elevated in relation to the indices for logical thinking and problem-solving. This signifies that was a greater number of scores away from the mean. Nonetheless, there was still a sturdy belief that coding enhances a student's creativity. The final skill in this set, collaboration, received the least support compared to the preceding three skills. The means were 1.89, 1.79, 2.10, and 1.67. The modes were all *strongly agree* aside from the high school subgroup and the standard deviations were 0.74, 0.96, 0.79, and 0.78, alluding to some responses veering away from the mean. Nevertheless, to be sure, these figures still refer to a relatively strong belief in coding's capacity to develop a student's collaborative skills.

The survey assessed perceptions of the development of general skills because they significantly impact student performance in different subjects, the acquisition of STEM occupations, and the consequent diversification of the STEM fields. When asked if coding instruction helps students perform more effectively in different courses, responses from the four different subgroups yielded means of 1.85, 1.87, 2.08, and 1.67 and modes of *strongly agree* except for the high school group, which primarily chose *somewhat agree*. The standard deviations of 0.85, 1.04, 0.70, and 0.83 signify the presence of responses above and below the mean and given that they are all either at the 1.0 mark or relatively close to it, it does represent a diverse set of responses. However, the means and modes indicate there is general agreement that coding facilitates student achievement in different courses.

Subsequent items focused on coding's impact on the acquisition of technology-based and non-technology-specific positions. The means were 1.42, 1.42, 1.62, and 1.15, *strongly agree* was the unanimous mode, and the standard deviations were 0.63, 0.60, 0.75, and 0.36. These figures represent a definite assertion that coding instruction will help students obtain positions of employment in the technology sector. The multiple cluster group, which contains some technology and CS teachers, unsurprisingly were the most bullish about coding's capacity to help students earn technology related positions. Responses were not as resolute, but still relatively firm, when evaluating coding's impact on the prospect of earning a position beyond the technology sector. This contention is evinced by means of 1.89, 2.18, 2.38, and 1.67 and modes of *strongly agree,* except for the high school subgroup's principal selection of somewhat agree.

The standard deviations 0.90, 1.01, 0.67, and 0.92. The high school subdivision had the lowest standard deviation which indicates more of participants' responses were closer to the mean. The middle school's mean was in close proximity to the response choice of *somewhat agree*, but its standard deviation represented a fair divergence away from the mean. Ultimately, the totality of responses indicate some agreement with coding's ability to equip students with the skills needed to secure positions in non-technology related sectors, but the respondents across all four subgroups perceive coding instruction to be more relevant for technology driven occupations.

The survey furthermore measured perceptions regarding coding instruction's effect on advancing social justice ends. Queries were posed asking about coding's ability to taper the disproportionality between male and female and between different ethnic groups' representation in STEM occupations. The means for the former item were 1.8, 1.58, 1.95, and 1.52 with modes of *strongly agree*, except for the high school group's chief selection of *somewhat agree*. The standard deviations were 0.91, 0.64, 0.76, and 0.75. All four groups, albeit the high school subdivision's slightly elevated measures, firmly indicate their belief in coding's potential to curtail the divide in representation between genders in STEM occupations. The high school group, relatively speaking, was a bit less enthusiastic about coding's ability to assist students in these different areas. As for coding's influence on diminishing the imparity in representation between different ethnic groups, the means of 1.82, 1.71, 2.0, and 1.63 and modes of *strongly agree*, except for the high school subdivision's mode of *somewhat agree*, depicted comparable perceptions to coding's capacity to diversify representation among females. The standard deviations of 1.02, 0.80, 0.79, and 0.84 once again signified some variation of scores away from the mean, but the indices suggest coding can advance this imperative social justice end.

Analysis of Teachers in Different School Levels and Administrators. The second research question yielded three separate constructs that evaluated the five subgroups' perceptions on coding instruction's impact on academic achievement, occupational preparedness, and diversifying the STEM academic and occupational fields.

Analysis of Second Construct. Five items, including perceptions on whether coding develops a student's logical thinking, problem solving, collaboration, performance in different subjects, and engagement in school, comprised this construct. The Cronbach's alpha was.83, as indicated in Table 17, and all the corrected item-total correlations were above 0.2 suggesting a strong interrelationship between the items. Furthermore, the alpha value would not have increased with the removal of any of the items.

Table 17

Construct 2—Perceptions of Coding's Impact on Academic Achievement

Survey Item	Response Option	α
		.83
Coding develops a student's logical thinking ability.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
Coding develops a student's problem solving ability.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
Coding develops a student's collaboration skills.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
Learning coding can help a student perform better in different subjects.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
Learning coding can helps a student be more engaged in school.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	

Means for the first two questions were robust (see Table 18), and the others remained quite solid as well, suggesting the overall belief that coding has the capacity to assist students in these five different areas and consequently improve student academic achievement. The highest mean, at 2.01, belonged to the item focused on student engagement. Although the figure demonstrated some agreement, it was somewhat surprising to note that coding was not perceived to assist as much with student engagement. Generally, coding takes place in collaborative groups in the classroom with the use of technological means, which can be argued to enhance engagement. Further analysis is necessary identify why this item received less support. The standard deviations ranged from 0.61 to 0.85, which portrays a modicum of variance away from the mean; however, this divergence was less with the questions on logical thinking and problem-solving abilities. These two skills are generally perceived to be connected to coding, which was highlighted by the data as well.

Table 18

Descriptive Statistics for Construct 2

Survey Item	*M*	*SD*	*n*
Coding develops a student's logical thinking ability.	1.44	0.67	186
Coding develops a student's problem solving ability.	1.4	0.61	186
Coding develops a student's collaboration skills.	1.85	0.81	186
Learning coding can help a student perform better in different subjects.	1.85	0.84	186
Learning coding can help a student be more engaged in school.	2.01	0.85	186

Table 19 underscores the descriptive statistics for the subgroups. The means ranged from 1.44 to 1.89, and the multiple cluster group, much like the first construct's results, had the lowest mean indicating their support for or inclination toward coding. The other groups also believed in coding's ability to develop various skills and improve students' achievement in the classroom. The elementary and middle school levels, as well as the administrators, were close in their perceptions. High school participants' scores were slightly higher, resulting in a mean of 1.89. Although the discrepancy is not significant, additional research would be necessary to understand why their perceptions were not as enthusiastic as the other groups. This pattern was also noted in the previous analyses as well. Standard deviations ranged from 0.49 to 0.70, which indicates some variance from the mean; however, the general data demonstrate relatively strong consistency. The overarching sentiment was that coding inculcates these various skills within students, positively impacting their achievement in different courses.

Table 19

Descriptive Statistics for Construct 2's Subgroups

Subgroup	*M*	*SD*	*n*
Elementary	1.75	0.60	54
Middle	1.71	0.70	35
High	1.89	0.49	34
Multiple cluster	1.44	0.50	25
Administration	1.67	0.56	38
Total	1.71	0.59	186

Inferential statistics, as noted in Table 20, provided additional data to determine whether there was a statistically significant difference in means between the five subgroups. Assumptions such as normality and homogeneity of variance were assessed, and, because they were violated, it is imperative to read the data with caution. The data did not have missing values, were based on 186 participants' responses, and were organized into five groups. The elementary subgroup had 54 participants (29.03%), the middle school subdivision had 35 (18.81%), the high school participants were 34 in number (18.27%), the multiple cluster subsection had 25 respondents (13.44%), and there were 38 administrators (20.43%). The response choices for construct 2, which evaluated participants' perceptions regarding coding instruction's impact on student academic achievement, ranged from 1 to 3.4 (M = 1.71, SD = 0.59). The Welch test's p value of.024 indicates there is at least one statistically significant difference in means between groups: Elementary (M = 1.75, SD = 0.60), Middle (M = 1.71, SD = 0.70), High (M = 1.89, SD = 0.49), Multiple Cluster (M = 1.44, SD = 0.50), and Administrator (M = 1.67, SD = 0.56), Welch's F (4, 83.26) = 2.97, p = .024, η_2 = .05.

Table 20

Robust Tests of Equality of Means for Construct 2

	Statistic	*df1*	*df2*	*p*	η_2
Welch	2.97	4	83.26	0.024	0.05

Note. df = degrees of freedom.

Post-hoc tests using Games-Howell (see Table 21) demonstrated the high school group was significantly different than the multiple cluster group (p =.010). The only other point-wise comparison that came slightly close to the .05 threshold was the elementary and multiple cluster group (p = .138). All other comparisons were significantly above the .05 p value threshold, which indicates a lack of a statistically significant difference between the means of the groups. Based on Cohen's guideline to interpret the effect size (eta squared or η_2), the point estimate value of .05 was deemed to be between small and moderate ($r2$ = .01 is small; $r2$ = .09 is medium; $r2$ = .25 is large). This indicates the subgroupings factored into the respondents' perceptions of coding's impact on academic achievement. Ultimately, these data suggest other than the statistically significant difference in means noted above, all other groups' perceptions related to coding's impact on academic achievement were relatively the same. This does not indicate their perceptions on the second construct are identical; rather, it denotes a lack of a statistically significant difference.

Table 21

Post Hoc Results for Construct 2 (Games-Howell)

Group	*M*	*SD*	Comparison Group	Mean Difference	*p*
Elementary	1.75	0.60	Middle	0.04	0.999
			High	-0.14	0.752
			Multiple Cluster	0.31	0.138
			Administrators	0.07	0.973
Middle	1.71	0.70	Elementary	-0.04	0.999
			High	-0.17	0.726
			Multiple Cluster	0.27	0.423
			Administrator	0.03	0.999
High	1.89	0.49	Elementary	0.14	0.752
			Middle	0.18	0.726
			Multiple Cluster	0.45	0.010
			Administrator	0.21	0.413
Multiple cluster	1.44	0.50	Elementary	-0.31	0.138
			Middle	-0.27	0.423
			High	-0.45	0.010
			Administrator	-0.23	0.426
Administrator	1.67	0.56	Elementary	-0.07	0.973
			Middle	-0.03	0.999
			High	-0.21	0.413
			Multiple Cluster	0.23	0.426
Total	1.71	0.59			

Analysis of Third Construct. The third construct focused on assessing perceptions on coding's impact on occupational preparedness as demonstrated in Table 22, specifically asking if coding in K–12 helps better prepare students to earn positions in technology-specific positions and in fields beyond this sector, such as in medicine and government. Furthermore, the survey probed whether coding is considered as an important 21st century literacy. This construct's Cronbach's alpha was .75, thus the questions were sufficiently interconnected. Furthermore, all three items' corrected item-total correlations were above .2, which also supports the contention that the items were connected. Removing items would not have yielded a high Cronbach's alpha.

Table 22

Construct 3—Perceptions of Coding's Impact on Occupational Preparedness

Survey Item	Response Option	α
		.75
Coding is an important 21st century literacy.	1= Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
Learning coding in K–12 will better prepare students to earn positions in technology-specific fields.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	
Learning coding in K–12 will better prepare students to earn positions in non-technology specific fields, such as medicine, retail, and government.	1 = Strongly Agree; 2 = Somewhat Agree; 3 = Neither Agree nor Disagree; 4 = Somewhat Disagree; 5 = Strongly Disagree	

Table 23 indicates means for the first two items were stout as suggested by the means of 1.63 and 1.44. It was not unforeseen that participants believed coding instruction would better prepare students to obtain technology related positions. This was also noted in every analysis of subgroups. Participants also predominantly perceived coding to be a significant 21st century literacy. However, the item probing coding's impact on non-technology related positions received a tepid response in comparison. Although the mean across all 186 individuals for this item was still relatively strong, it was not as affirmative in comparison to the first queries. This piece of data was in consonance with every analysis of subgroup sets; participants generally believed coding assists less in the acquisition of non-technology related positions. The standard deviations ranged from 0.63 to 0.91, thus some variance from the mean was noted in the responses. Nevertheless, these figures stemming from 186 people's perceptions do indicate firm support for the items in this construct.

Table 23

Descriptive Statistics for Construct 3

Survey Item	*M*	*SD*	*n*
Coding is an important 21st century literacy.	1.63	0.81	186
Learning coding in K–12 will better prepare students to earn positions in technology-specific fields.	1.44	0.63	186
Learning coding in K–12 will better prepare students to earn positions in non-technology specific fields, such as medicine, retail, and government.	2.04	0.91	186

Data on the five subgroups, as indicated in Table 24, yielded significant, yet anticipated data. All subgroups' means indicated support for the three items in the construct, with the multiple cluster group once again asserting the greatest support for coding due to its membership. To reiterate, this group contains some teachers who instruct technology or CS courses. All other groups ranged in means from 1.67 to 1.86, with the high school group exhibiting the highest average. In fact, the high school's generated data were quite similar to the second construct's high school data. Although the 1.86 mean figure was not significantly above the others, it is nevertheless noteworthy that their perception regarding coding's capacity to assist student wanes, albeit slightly, relative to other subgroups. Every subgroup's standard deviation showed a modicum of digression from the mean, but these figures generally demonstrated a relative clustering around the mean, signifying the belief that coding instruction in K–12 does assist in occupational preparedness.

Table 24

Descriptive Statistics for Construct 3's Subgroups

Subgroup	*M*	*SD*	*n*
Elementary	1.67	0.66	54
Middle	1.80	0.75	35
High	1.86	0.51	34
Multiple Cluster	1.39	0.59	25
Administrators	1.72	0.63	38
Total	1.70	0.65	186

Inferential statistics was also employed to further understand the third construct's data as to whether there was a statistically significant difference in means between the five subgroups. Assumptions such as normality and homogeneity of variance were assessed and because they were violated, it is imperative to read the data with caution. The data did not have any missing values and were based on 186 participants' responses. Furthermore, data were organized into five groups. The elementary subgroup had 54 participants (29.03%), the middle school subdivision had 35 (18.81%), high school participants were 34 in number (18.27%), the multiple cluster subsection had 25 respondents (13.44%), and there were 38 administrators (20.43%). The response choices for Construct 3, which evaluated participants' perceptions of coding instruction's effect on occupational preparedness, ranged from 1 to 3.67 (M = 1.70, SD = 0.65). Table 25 highlights the Welch test's p value of .034, which indicates the null hypothesis is rejected and there is at least one statistically significant difference in means between groups: Elementary (M = 1.67, SD = 0.66), Middle (M = 1.80, SD = 0.75), High (M = 1.86, SD = 0.51), Multiple Cluster (M = 1.39, SD = 0.59), and Administrator (M = 1.72, SD = 0.63), Welch's F (4, 82.85) = 2.75, p =.034, η_2 =.05.

Table 25

Robust Tests of Equality of Means for Construct 3

	Statistic	*df1*	*df2*	*p*	η_2
Welch	2.75	4	82.85	0.034	0.05

Note. df = degrees of freedom.

Table 26 demonstrates the data from the post-hoc tests using Games-Howell, which suggested the high school group was significantly different than the multiple cluster group (p =.018). The only other point-wise comparison that came slightly close to the .05 threshold was the middle and multiple cluster groups (p =.134). All other comparisons were significantly above the .05 p value threshold, which indicates a lack of a statistically significant difference between the means of the groups. Based on Cohen's guideline to interpret the effect size (eta squared or η_2), as stated by P. Ruengvirayudh (personal communication, March 13, 2022), the point estimate value of .048 was deemed to be between small and moderate ($r2$ = .01 is small; $r2$ = .09 is medium; $r2$= .25 is large). This indicates the subgroupings factored into respondents' perceptions of coding's impact on occupational preparedness. Ultimately, these data suggest, other than the statistically significant difference in the means noted above, all other groups' perceptions regarding this construct were relatively the same. However, this does not indicate their perceptions are not different; rather, it signifies a lack of a statistically significant difference between the subdivisions.

Table 26

Post Hoc Results for Construct 3 (Games-Howell)

Group	*M*	*SD*	Comparison Group	Mean Difference	*p*
Elementary	1.67	0.66	Middle	-0.13	0.924
			High	-0.19	0.554
			Multiple cluster	0.29	0.317
			Administrators	-0.05	0.997
Middle	1.80	0.75	Elementary	0.13	0.924
			High	-0.06	0.994
			Multiple cluster	0.41	0.134
			Administrator	0.08	0.987
High	1.86	0.51	Elementary	0.19	0.554
			Middle	0.06	0.994
			Multiple cluster	0.48	0.018
			Administrator	0.14	0.820
Multiple cluster	1.39	0.59	Elementary	-0.29	0.317
			Middle	-0.41	0.134
			High	-0.48	0.018
			Administrator	-0.33	0.221
Administrator	1.72	0.63	Elementary	0.05	0.997
			Middle	-0.08	0.987
			High	-0.14	0.820
			Multiple cluster	0.33	0.221
Total	1.70	0.65			

Note: Sig.= significance.

Analysis of Fourth Construct. This construct gauged perceptions on coding's impact on STEM representation in academics and in the workforce. Table 27 highlights the questions urging participants to opine on whether coding can bridge the divide in K–12 STEM interest *and* representation in STEM occupations between males and females and among ethnic groups.

Table 27

Construct 4—Perceptions of Coding's Impact on STEM Representation

Survey Item	Response Option	*α*
		.94
Coding can help bridge the gap in K–12 STEM interest between males and females.	1= Strongly Agree; 2=Somewhat Agree; 3=Neither Agree nor Disagree; 4=Somewhat Disagree; 5=Strongly Disagree	
Coding can help bridge the gap in K–12 STEM interest between different ethnic groups.	1= Strongly Agree; 2=Somewhat Agree; 3=Neither Agree nor Disagree; 4=Somewhat Disagree; 5=Strongly Disagree	
Exposure to coding in K–12 schooling will help bridge the gap in STEM occupations held by males and females.	1= Strongly Agree; 2=Somewhat Agree; 3=Neither Agree nor Disagree; 4=Somewhat Disagree; 5=Strongly Disagree	
Exposure to coding in K–12 schooling will help bridge the gap in representation amongst different ethnic groups in STEM occupations.	1= Strongly Agree; 2=Somewhat Agree; 3=Neither Agree nor Disagree; 4=Somewhat Disagree; 5=Strongly Disagree	

The .94 Cronbach's alpha and the corrected item total correlations, which all measured above 0.2, demonstrated significant interrelatedness between the items. The Cronbach's alpha would not have increased with the removal of any of the items. Thus, the deductions drawn from the statistics would adequately explain perceptions regarding coding's impact on STEM representation.

The means for all four questions, which are represented in Table 28, were comparable ranging from 1.73 to 1.95. For both sets of items, participants believed coding better assisted females' interest and representation in STEM fields compared to different ethnic groups. To be sure, the difference was minimal; however, it is curious to know what the difference was due to. Additional research would be necessary to properly explicate this distinction. The standard deviations ranged from 0.78 to 0.93, which signify some digression from the means. The standard deviations for the two items regarding ethnic groups were also higher relative to the females' data, which signifies greater variance from the mean. Yet, the data provide fodder to participants' contention that coding enhances interest in STEM among females and different ethnic groups and it diversifies representation in STEM academics and occupations.

Table 28

Descriptive Statistics for Construct 4

Survey Item	*M*	*SD*	*n*
Coding can help bridge the gap in K–12 STEM interest between males and females.	1.82	0.85	186
Coding can help bridge the gap in K–12 STEM interest between different ethnic groups.	1.95	0.93	186
Exposure to coding in K–12 schooling will help bridge the gap in STEM occupations held by males and females.	1.73	0.78	186
Exposure to coding in K–12 schooling will help bridge the gap in representation amongst different ethnic groups in STEM occupations.	1.81	0.87	186

The assessment of the five subgroups yielded the same outcomes as the other two sets of data in this construct (see Table 29). The elementary, middle, and administrator responses were analogous. The multiple cluster group's data produced the lowest mean indicating the strongest agreement in coding's ability to affect positive change, and the high school group showed some agreement with the statements, but they were not nearly as passionate. Because this overarching result has been noted with all three constructs as part the second research question, additional analysis on the rationales for these responses would need to be vetted to offer an accurate explanation.

Table 29

Descriptive Statistics for Construct 4's Subgroups

Subgroup	*M*	*SD*	*n*
Elementary	1.86	0.92	54
Middle	1.72	0.76	35
High	2.03	0.72	34
Multiple clusters	1.57	0.75	25
Administrators	1.87	0.66	38
Total	1.83	0.79	186

Inferential statistics were also employed to further understand the fourth construct's data as to whether there was a statistically significant difference in means between the five subgroups. Assumptions such as normality and homogeneity of variance were assessed, and, because they were violated, it is imperative to read the data with caution. The data did not have any missing values and it was based on 186 participants' responses. Furthermore, data were organized into five groups. The elementary subgroup had 54 participants (29.03%), the middle school subdivision had 35 (18.81%), high school participants were 34 in number (18.27%), the multiple cluster subsection had 25 respondents (13.44%), and there were 38 administrators (20.43%). Response choices for Construct 4, which gauged participants' perceptions of coding instruction's impression on STEM representation, ranged from 1 to 4.5 (M = 1.83, SD = 0.79). Table 30 highlights the Welch test's p value of .188, which indicates there is no statistically significant difference in means between groups: Elementary (M = 1.86, SD = 0.92), Middle (M = 1.72, SD = 0.76), High (M = 2.03, SD = 0.72), Multiple Cluster (M = 1.57, SD = 0.75), and Administrator (M = 1.87, SD = 0.66), Welch's F (4, 83.31) = 1.58, p =.188, η_2 =.031. Because the p value did not meet the .05 threshold, a post-hoc test was not conducted to determine significance in point-wise comparisons. The data did not show a statistically significant difference in means regarding coding's influence on diversifying STEM representation.

Table 30

Robust Tests of Equality of Means for Construct 4

	Statistic	*M (avg.)*	*SD (avg.)*	*df1*	*df2*	*p*	η_2
Welch	1.58	1.83	0.79	4	83.31	0.188	0.031

Note. df = degrees of freedom.

Second Research Question's Conclusion

Data stemming from the total sample population offered the compelling perception that coding develops various skills such as problem solving, logical thinking, and collaboration, it spurs interest in K–12 STEM subjects within females, males, and different ethnic groups, better positions them to attain technology and to a lesser degree non-technology-related occupations, and it advances female and minority population representation in STEM occupations.

The analysis of data stemming from the different subgroups of teachers, administrators, and participants with multiple roles clearly demonstrated the general belief that coding assists in developing logical thinking, problem solving, creativity, and to a lesser degree, collaboration. These indices were identical with the subgroups of female and male respondents. An analysis of grade-level clusters depicted identical overarching results as well. Furthermore, subgroups of teachers, administrators, and participants with multiple roles believed coding can assist students in performing more effectively in different courses, it better positions them to secure technology and non-technology related positions, and it has the potential to enhance female and different ethnic groups' representation in STEM occupations. Females and males, too, believed in coding's capacity to assist students in different classes and to better position them to secure technology related positions. However, they were not as effusive regarding its potential to assist students in being employed in non-technology related sectors. These sentiments were shared by different grade-levels clusters. Furthermore, females did believe coding can enrich female and different ethnic groups' representation in STEM occupations, but the data from male responses signified a lackluster perception regarding coding's capacity to achieve this. The scrutiny of the grade-level subdivisions revealed analogous results. All four clusters, except for the marginally less fervent perceptions among participants from high school, affirmed coding instruction's ability to enhance representation among females and different ethnic groups in STEM occupations.

The final analysis of the five subgroups indicated a firm belief in coding's ability to develop various skills, enhance student achievement in the classroom, and assist with engagement. The multiple cluster group, which contains participants who are CS and technology teachers, were the most enthusiastic toward these survey items; however, the other subgroups' perceptions were not far behind. Furthermore, all subgroups believed in coding instruction's capacity to enhance student occupational preparedness, especially as it relates to technology specific positions. The multiple cluster group was the most effusive and the high school group's responses waned in comparison to the others. These two points of data mimicked the figures noted in the second construct as well. Construct 4 appraised the effect of coding on enhancing female and ethnic group interest in STEM in school and its influence upon STEM representation. The data suggest the perception that coding furthers females' and ethnic populations' interest in STEM in school, and it expands representation in STEM occupations. The multiple cluster group was keen on this potential, and the high school group regarded it with some reservation, although the data generally demonstrated some agreement.

The inferential statistics from the first construct additionally yielded valuable data. The difference in means between the multiple cluster and elementary group were statistically significant (p = .005). Furthermore, another statistically significant difference in means was noted between the multiple cluster and administrator subgroups (p = .009). This difference was slightly less relative to the difference between the multiple cluster and elementary group, but sufficient nonetheless. The difference in means between the multiple cluster and high school group was also noteworthy with a p value of .08, but it did not cross the threshold for significance. Nevertheless, this piece of data shows some difference between these two groups. Ultimately, the data demonstrate the multiple cluster group's understanding of coding is superior, which may be due to the group's membership containing participants who teach CS and technology. Surely, instructing these subjects would indicate those participants have a firmer understanding of coding. If other members of the subgroups would be exposed to CS, technology, and coding, there understanding would enhance as well and they would thus be better suited to facilitate its instruction in the classroom.

The second construct's post-hoc test also demonstrated a statistically significant difference between the high school and multiple cluster subgroups, as indicated by the p value of 0.010 and F ratio of 2.97. There were no other statistically significant differences in means; however, that is not to suggest all perceptions related to coding's impact on academic achievement were identical. Subtle differences may exist, yet the p value's .05 threshold for significance, which is meant to limit the possibility of the difference being due to random and unsystematic reasons (P. Ruengvirayudh, personal communication, March 12, 2022), was not met by other point-wise comparisons.

The third construct bore similar results to the second, as indicated by a statistically significant difference of p = .018 between the high school and multiple cluster subgroups. Other than a *p* value of .134 between administrators and the multiple cluster group, no other point-wise comparison was remotely close in proximity to the .05 threshold. These findings assert there was no statistically significant difference between the other subgroups in the analysis. These data points are reasonable because the multiple cluster and the high school group were typically outliers, as the former was considerably more ebullient in their perceptions of coding's positive impact and the latter was less enthusiastic.

The fourth construct was dissimilar relative to the first three because it presented no statistically significant differences in means, as demonstrated by the Welch *F* ratio of 1.54 and *p* value of .188. Consequently, a post-hoc test was not conducted because there would be no statistically significant differences in means with point-wise comparisons. This does not suggest there is no difference; indeed, there may very well be; however, there was no evidence of a statistically significant difference. This may indicate the subgroups' perceptions for this construct were relatively harmonious.

Third Research Question

A final line of inquiry presented itself after I developed the original survey instrumentation. Papert's constructionism, which is in line with coding instruction in the classroom, was a focal point of my literature review, and I wanted to examine whether practitioners believed it was a viable framework for use in the classroom. Therefore, I asked the following question: How do K–12 ADLA Catholic teachers', administrators', and STEM directors'/coordinators' epistemology and pedagogy impact their inclination to integrate the tenets of constructionism and coding?

Survey Results

The following section will provide a robust discussion explaining the results of the various analyses conducted as part of the third research question, including, but not limited to, an assessment of the entire set of data and an analysis of data based on the position of the respondents, their stated gender, and grade level of instruction,

Analysis of Total Sample Population. Educators possess varying pedagogical beliefs and use practices that are in line with them. Indeed, Chan and Elliot (2004) stated, "The conceptions about teaching and learning refer to the beliefs held by teachers about their preferred ways of teaching and learning. These include the meaning of teaching and learning and the roles of teacher and pupils" (p. 819). Consequently, as noted in Table 31, I posed various questions to ascertain the educators' pedagogical and epistemological beliefs and determine whether there is a linkage between their teaching philosophies and a potential adherence to constructionism's focus on technology and student-driven collaboration and creation.

Table 31

Total Sample Population's Descriptive Statistics for Third Research Question's Items

Survey Item	*M*	*SD*	Mode
Direct instruction is an effective means to student learning.	2.33	1.04	2
Student collaboration and communication are important components of learning.	1.23	0.52	1
Learning means remembering what the teacher has taught.	3.11	1.23	2
Teaching is presenting or explaining the subject matter.	2.52	1.15	2
Technology should be a focal tool in the classroom.	2.34	1.06	2
The focus of teaching is to help students construct knowledge from their learning experiences instead of imparting knowledge.	1.68	0.79	1
Students should have greater control over their learning process.	1.86	0.79	2
Constructionism is possible to apply in the classroom.	1.53	0.52	2

Direct instruction is a traditional form of teaching whereby teachers lecture or provide information to students. Although this form of teaching possesses merit and value, there has been a recent shift veering away from the teacher's central role in the classroom. There are many other modalities of teaching as well, and to assess whether constructionism is feasible to apply in Catholic school classrooms in the ADLA, it is imperative to gauge the respondents' pedagogical and epistemological philosophies. As such, I inquired into what educators in ADLA Catholic schools believe regarding direct instruction and other forms of pedagogy. I posed a question asking respondents to order their preferred methods of teaching based on six choices, which included direct instruction, student cooperative and collaborative learning, direct instruction followed by student application, inquiry-based learning, flipped classroom, and kinesthetic learning. Thirty-two participants chose direct instruction as their most employed method of instruction, 22 selected student cooperative and collaborative learning, 22 elected direct instruction followed by student application, 20 preferred inquiry-based learning, 26 chose a flipped classroom, and 62 picked kinesthetic learning. In regard to the second most preferred manner of teaching, 26 selected direct instruction, 93 elected student cooperative and collaborative learning, 49 chose direct instruction followed by student application, 14 preferred inquiry-based learning, two chose a flipped classroom, while none picked kinesthetic learning. Participants' third most applied practice of instruction was direct instruction with 102, 36 chose collaborative and cooperative learning, 26 elected direct instruction followed by student application, 12 chose inquiry-based learning, 10 selected a flipped classroom, and none picked kinesthetic learning. These data suggest direct instruction, whether on its own or followed by student application, plays a prominent role in teacher instruction. However, student cooperative and collaborative learning is also a method frequently used by the respondents. The outlier was kinesthetic learning as 62 respondents chose it as their chief method of instruction. The survey was available to all K–12 teachers and administrator and kinesthetic learning is a modality ordinarily applied in lower grade levels. The data from this question is noteworthy as direct instruction on its own does not necessarily lend itself to constructionism; however, if it is followed by student application with the use of technology, then it is feasible. A teacher may explain the content and subsequently allow students to apply it as they code. Moreover, cooperative and collaborative learning is a key tenet of constructionism and employing this strategy during the coding process enables students to learn with and from each other.

An additional question from the survey prompted participants to opine on whether direct instruction is an effective medium for teaching. The mean for this item was a 2.33 and the mode was *somewhat agree* (2), and the standard deviation was 1.04. Some responses *strongly agreed* with the importance of direct instruction while others *neither agreed nor disagreed*. However, the predominant response choice is consistent with the shift away from direct instruction. The statement attempting to assess educators' position on whether teaching is presenting or explaining the subject matter generated a mean of 2.52, a standard deviation of 1.15, and mode of *somewhat agree* (2). Much like the previous question regarding direct instruction, this survey item gauged whether the teacher is the central figure in the classroom. There is some agreement with the statement, as suggested by the mean and mode indices; however, there is a large standard deviation indicating variance away from the mean. Although the teacher presenting or explaining the subject matter does not preclude the integration of coding and constructionism in the classroom, it is important for the teacher to allow for collaborative and creative work. Moreover, the survey item prompting participants to respond to whether learning is remembering what the teacher has taught, the mean was greater at 3.11, the standard deviation was also more considerable at 1.23, but the mode was still *somewhat agree* (2). Interpreting these data was challenging because the mean showed no firm stance either way, yet the mode was *somewhat agree*, which does not necessarily line well with constructionist philosophy. The standard deviation presented a variance in scores away from the mean, which explains the full 1-point higher mean compared to the mode. In totality, this perception regarding the teacher's central role may conflict with integrating the constructionist paradigm.

The rest of the items in this block had smaller means and standard deviations. For instance, the item on the importance of student collaboration and communication produced a mean of 1.23, a standard deviation of 0.52, and a mode of *strongly agree* (1). Furthermore, the item that gauged perceptions on the importance of facilitating student construction of knowledge from their previous experiences instead of imparting knowledge yielded a mean of 1.68, a standard deviation of 0.79, and a mode of *strongly agree* (1). Teachers also suggested students should have greater control over the learning process, which was signaled by a mean of 1.86, a standard deviation of 0.79, and a mode of *somewhat agree* (2). These data demonstrated educators' belief that students hold a greater primacy in the dynamic of the classroom, which will enhance the possibility of integrating constructionism and coding in the classroom. There, however, needs to be a reconciliation between teacher- and student-centered strategies. The data demonstrated the perception that certain teacher-centered strategies are important, yet participants *somewhat agreed* with the need to give students a greater role in the classroom, which is a more student-centered orientation. Potentially, an effective combination of strategies may be possible to employ while integrating constructionism and coding.

Responses for the two items related to technology and constructionism were not completely in consonance. The question asking whether technology should be a focal tool in the classroom produced a mean of 2.34, a standard deviation of 1.06, and a mode of *somewhat agree* (2); however, when asked if constructionism is feasible to institute in the classroom, participants' responses generated a mean of 1.53, a standard deviation of 0.52, and mode of possible, but difficult to do (2). Of note, the question on constructionism was assessed with a different scale, which ranged from 1 to 3 (easily possible, possible but difficult to do, and not possible). Although both modes are comparable because they indicate some agreement or possibility, there are discrepancies between the means and standard deviations. Before participants responded to the question on constructionism, they were provided with a brief description of it, which included the use of technology to allow students to create products collaboratively. The mean score of 1.52 suggests some agreement with the statement on constructionism; however, participants were not as willing to concede to the importance of technology as a focal tool in the classroom. This does not suggest educators do not believe in the utility of technology; however, data for this item were not as demonstrative as the one on constructionism. Because constructionism is based on the use of technology, if teachers are relatively tepid on its use in the classroom, it would naturally be more difficult to apply the paradigm in the classroom. One possible reason why teachers do not consider technology as a focal tool in the classroom is the potential for its misuse. As an educator, I can vouch for the inherent challenges involved when using technology in the classroom. For instance, students may be distracted by games and social media, which detracts from the classroom experience and may dissuade educators from relying on it as a central pedagogical tool. Additionally, educators may not necessarily feel comfortable integrating technology because of personal discomfort or lack of experience with it. Such deterring reasons can be ameliorated; however, the educator must genuinely choose to use technology in the classroom for constructionism to take root and blossom.

Analysis of Subgroups Based on Position. In addition to assessing the entire population's perceptions on the benefits of coding instruction, this study isolated and evaluated data based on subgroupings of teachers, administrators, and participants with multiple roles (see Table 32). Participants' epistemological and pedagogical foundations evidenced whether they were inclined to integrate constructionism and coding into their classroom instruction. For instance, Item 54 asked whether direct instruction is an effective means to student learning. The teachers' responses returned a mean of 2.29, a mode of *somewhat agree*, and a standard deviation of 0.98. The administrators' mean was 2.39, the mode was *somewhat agree*, and the standard deviation was 1.15. The replies of the group with multiple roles produced a mean 2.34, a mode of *somewhat agree*, and a standard deviation of 1.12. The means and modes from all three groups demonstrate the perception that direct instruction is of somewhat utility. The relatively large standard deviations also demonstrate there were some responses that did not cluster around the mean; therefore, there was a relatively wide range of responses here. Direct instruction does not necessarily harmonize with constructionist philosophy, yet if it is followed by collaborative work, then it is still possible to integrate it.

Table 32

Descriptive Statistics on Epistemology and Constructionism (Based on Participants' Position)

	Teacher			Admin			MR		
Survey Item	*M*	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode
Direct instruction is an effective means to student learning.	2.29	0.98	2	2.39	1.15	2	2.34	1.12	2
Student collaboration and communication are important components of learning.	1.30	0.62	1	1.16	0.37	1	1.08	0.27	1
Learning means remembering what the teacher has taught.	2.96	1.28	2	3.45	1.08	4	3.16	1.20	2
Teaching is presenting or explaining the subject matter.	2.54	1.16	2	2.5	1.13	2	2.53	1.18	2
Technology should be a focal tool in the classroom.	2.31	1.08	2	2.47	0.98	2	2.29	1.09	2
The focus of teaching is to help students construct knowledge from their learning experience instead of imparting knowledge.	1.69	0.78	1	1.58	0.79	1	1.74	0.83	1
This framework [constructionism] is possible to apply in the classroom.	1.58	0.53	2	1.53	0.51	2	1.34	0.48	1

Note. MR = multiple roles.

The next item that is crucial in determining whether their epistemology and pedagogy are in line with constructionism and the integration of coding prompted participants to identify whether student collaboration and communication are to important components of learning. The means were 1.30, 1.16, and 1.08, respectively, and the modes were all *strongly agree*. The standard deviations of 0.62, 0.37, and 0.27 signal a strong convergence of responses around the means. These overall figures signify the stout perception that collaboration and communication are integral elements of student learning. The survey also attempted to evaluate participants' stance on a more teacher-centered classroom environment. As such, the survey prompted a response on whether learning means remembering what the teacher has taught and if teaching is presenting or explaining the subject matter. For the former question, the respective means were 2.96, 3.45, and 3.16. The modes were *somewhat agree* (the second answer choice on the Likert scale) for the teacher and multiple roles subgroups, and *somewhat disagree* (the fourth response option on the Likert scale) for administrators. The standard deviations were 1.28, 1.08, and 1.20. In totality, the data signifies a relatively tepid belief in this statement. Although the means are closer around the third response choice, which states *neither agree nor disagree*, the modes of *somewhat agree* for two groups show a healthy number of participants somewhat perceive it as being true. All three standard deviations are greater than 1, which indicates a formidable number of responses diverging away from the mean; however, there is enough sentiment in relative favor of this item to question whether participants would integrate constructionism in the classroom. The statistical outlier is the administrators' mode of *somewhat disagree*, which may signify their inclination away from a teacher-centered environment.

The item that assessed whether teaching is presenting or explaining the subject matter, generated means of 2.54, 2.5, and 2.53, modes of solely *somewhat agree*, and standard deviations of 1.16, 1.13, and 1.18. These numbers indicate almost an identical position on this statement, which suggests participants somewhat believed teaching is highlighted by presenting or explaining the subject matter. All the means hovered between the second and third response choices, but the comparably high standard deviations signify responses did not cluster around the means. As such, participants' responses would have yielded some 1s (*strongly agree*) and 4s (*somewhat disagree*).

Because Papert's constructionism champions a student-centered environment through the use of technology, participants were queried as to whether it should be a focal tool in the classroom. The means were 2.31, 2.47, and 2.29, the modes were all *somewhat agree*, and the standard deviations were 1.08, 0.98, and 1.09. Once again, these figures demonstrate a parallel line of thinking among all three groups, which does somewhat consider technology to be a central tool in the classroom, but not as profoundly as one would expect given the current expanding use of technology in the classroom. To integrate coding and constructionism in the classroom, teachers and administrators would need to be fully committed to the use of technology. There are inhibiting factors such as cost and lack of teacher preparation to integrate technology consistently in the classroom; however, in a constructionist academic setting, technology is indispensable. To better prepare teachers to integrate technology in the classroom effectively, resources for development and training are crucial.

Additionally, a chief tenet of constructionism asserts that students should have the opportunity to construct knowledge from their learning experiences rather than information being imparted to them. As such, survey item 59 assessed this and the means for all three groups were 1.69, 1.58, and 1.74 while the modes were all *strongly agree*. Consequently, all three groups affirmed the importance of this point. The standard deviations of 0.78, 0.79, and 0.83 showed a relative convergence around the means. This overall perception is important because student creation of knowledge is a key tenet of constructionism, making it potentially more likely for its integration. The final item regarding constructionism briefly explained its components and directly asked whether this paradigm can be applied in the classroom. The means for this item were 1.58, 1.53, and 1.34. To reiterate, the scale for this question was different because it was rated with a score of 1–3, representing *easily possible* (1), *possible but difficult to do* (2), and *not possible* (3). The teacher and administrator subgroups' modes were possible but difficult to do, and the multiple roles subgroup generated a mode of easily possible. The means and modes for these two items indicate the general perception that constructionism is possible to apply in the classroom, although it will not be easy to do so. Data from this straightforward question are a firm signal of the potential to make Papert's framework a reality in the classroom.

Analysis of Subgroups Based on Gender. The subgroupings of females and males were also used to assess participants' perceptions of pedagogy and epistemology and its link to integrating constructionism and coding in the classroom. Constructionism champions a student-centered environment where students are given the platform to explore their creative and collaborative potentials through the creation of coded products. To best assess the potential to establish this environment in the classroom, the survey asked several questions including whether direct instruction is an effective means to student learning, which are highlighted in Table 33.

Table 33

Descriptive Statistics on Epistemology and Constructionism (Based on Gender)

Survey Item	Female			Male		
	M	*SD*	Mode	*M*	*SD*	Mode
Direct instruction is an effective means to student learning.	2.26	1.03	2	2.51	1.03	2
Student collaboration and communication are important components of learning.	1.21	0.51	1	1.28	0.55	1
Learning means remembering what the teacher has taught.	3.07	1.20	2	3.16	1.33	2
Teaching is presenting or explaining the subject matter.	2.54	1.15	2	2.42	1.12	2
Technology should be a focal tool in the classroom.	2.33	1.03	2	2.33	1.11	2
The focus of teaching is to help students construct knowledge from their learning experience instead of imparting knowledge.	1.67	0.79	1	1.65	0.78	1
This framework [constructionism] is possible to apply in the classroom.	1.54	0.51	2	1.49	0.55	1

Females' responses netted a mean of 2.26, a mode of *somewhat agree*, and a standard deviation of 1.03 while male responses returned a mean of 2.51, a mode of *somewhat agree*, and a standard deviation of 1.03 as well. This demonstrates both female and male participants deem direct instruction to possess some utility. The survey also asked whether learning means remembering what the teacher has taught. The survey asked this to assess whether there is any affirmation for this point because it deviates from constructionism's core tenants. The female data yielded a mean of 3.07, a mode of *somewhat agree*, and a standard deviation of 1.20; the male data produced a mean of 3.16, a mode of *somewhat agree*, and a standard deviation of 1.33. The means above 3 signify less agreement with this statement; however, the modes were still *somewhat agree* for both groups, and the standard deviations above 1.0 highlight a wider range of responses away from the mean. Therefore, although the modes are *somewhat agree*, there are considerable numbers of responses of *neither agree nor disagree* and *somewhat disagree*. Ultimately, this bodes relatively well for the prospect of integrating constructionism and coding in the classroom. Yet another question that opposed constructionism principles urged the respondent to rate the statement of whether teaching is presenting or explaining the subject matter. Female data created a mean of 2.54, a mode of *somewhat agree*, and a standard deviation of 1.15. The data from males resulted into a mean of 2.42, a mode of *somewhat agree*, and a standard deviation of 1.12. These data points demonstrate relatively tepid support. The larger standard deviations and the modes of *somewhat agree* underscore the reality that some responses shifted away from the mean; however, unlike the previous question's means, these ones are much closer to *somewhat agree*, which conflicts with constructionism's assertions.

There were questions or statements in line with constructionism, and the study assessed participants' pedagogical and epistemological foundations with an evaluation of these items as well. Statements on whether student collaboration and communication are important components of learning netted a female mean of 1.21, a mode of *strongly agree*, a standard deviation of 0.51 while the query on whether technology is a focal tool in the classroom produced a mean of 2.33, a mode of *somewhat agree* (2), and a standard deviation of 1.03. Male participants' responses to the collaboration and communication item created a mean of 1.28, a mode of *strongly agree*, and a standard deviation of 0.55; however, the question of whether technology is a focal tool in the classroom generated a larger mean of 2.33, a mode of *somewhat agree*, and a standard deviation of 1.10. The question on collaboration and communication generated a much more positive response among both subgroups, which would enhance the potential for integrating constructionism and coding; however, reactions to the item on technology serving as a focal tool were not as affirmatory. The means of around 2.0 and the modes of *somewhat agree* are noteworthy, but participants' convictions toward the use of technology need to most likely be stouter to incorporate its use in the classroom consistently. Another essential question about a core constructionist principle question assessed whether participants believe the focus of teaching is to help students construct knowledge from their learning experiences instead of imparting knowledge. Female responses generated a mean of 1.67, a mode of *strongly agree*, and a standard deviation of 0.79 and the male participants' data culminated into a mean of 1.65, a mode of *strongly agree*, and a standard deviation of 0.78. This set of data suggests both subgroups believe in this pedagogical and epistemological principle, which bodes well for integrating constructionism and coding in the classroom.

The final question posed to these two subgroups directly asked whether constructionism is possible to employ in the classroom. A brief summation of constructionism's precepts was provided before respondents were urged to answer this question. Keeping in mind that the scale for this question was rated with scores ranging from 1–3 (easily possible, possible but difficult to do, and not possible), the female data points produced a mean of 1.54, a mode of possible but difficult to do, and a standard deviation of 0.51. The male responses also created a strong mean of 1.49, a mode of easily possible, and a standard deviation of 0.55. The females' data points suggest respondents chose possible but difficult to do more than easily possible by a slight margin. Although the data are quite similar, males primarily chose *easily possible*, and females were a bit more reserved in their assessment. Although additional research is necessary to provide a proper reasoning for this difference, it may be attributed to the historically greater presence of males in the STEM fields. Ultimately, however, the responses underscore the possibility of incorporating constructionism in the classroom.

Analysis of Subgroups Based on Grade-Level Clusters. Data from these subdivisions, as emphasized in Table 34, were evaluated to determine whether their epistemological and pedagogical leanings enhance the potential for their integration of coding and constructionism in the classroom. Constructionism is one vehicle through which coding can be integrated into the classroom and as such, questions related to this framework were posed as well.

Table 34

Descriptive Statistics on Epistemology and Constructionism (Based on Level of Instruction)

Item	Elem			Mid			High			MC		
	M	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode	*M*	*SD*	Mode
Direct instruction is an effective means to student learning.	2.33	0.94	2	2.21	1.09	2	2.38	1.04	2	2.41	1.12	2
Student collaboration and communication are important components of learning.	1.22	0.46	1	1.16	0.32	1	1.28	0.56	1	1.30	0.82	1
Learning means remembering what the teacher has taught.	2.89	1.17	2	3.21	1.14	2	2.95	1.33	2	3.15	1.43	2
Teaching is presenting or explaining the subject matter.	2.42	1.10	2	2.47	1.27	2	2.77	1.11	2	2.37	1.11	2
Technology should be a focal tool in the classroom.	2.11	0.99	2	2.34	1.10	2	2.51	1.05	2	2.48	1.28	2
The focus of teaching is to help students construct knowledge from their learning experience instead of imparting knowledge.	1.69	0.72	2	1.61	0.68	1	1.85	0.87	2	1.63	0.88	1
This framework [constructionism] is possible to apply in the classroom.	1.67	0.51	2	1.42	0.50	1	1.62	0.54	2	1.26	0.45	1

Note. MC = multiple cluster.

The first set of questions queried epistemological beliefs and pedagogical strategies that run counter to constructionism's tenets to evaluate whether incorporating coding and constructionism in the classroom is feasible. Participants were asked whether direct instruction is an effective means to student learning and their responses generated means of 2.33, 2.21, 2.38, and 2.41 and a unanimous mode of *somewhat agree*. The standard deviations were 0.94, 1.09, 1.04, and 1.12; therefore, some divergence away from the means occurred. Yet, the means' figures of slightly above 2.0 and the modes demonstrate some agreement with direct instruction's utility in the classroom. Another item asked whether learning means remembering what the teacher taught. The means for this item were 2.89, 3.21, 2.95, 3.15 with a common mode of *somewhat agree*. The standard deviations were calculated as 1.17, 1.14, 1.33, and 1.43. Relative to the question focusing on direct instruction, this item generated greater means, but the same mode of *somewhat agree*. However, the standard deviations were also greater explaining the reason for the higher means while the modes remained the same. The divergence away from the mean indicates there were responses on both the front and end of the scales; however, because the means are closer to the end of the scale, a considerable number of the scores were found to be on the far end. An additional item analyzed perceptions on whether teaching is presenting or explaining a subject matter. The data produced means of 2.42, 2.47, 2.77, and 2.37 and an identical mode of *somewhat agree*. The standard deviations were all above 1.0 with 1.10, 1.27, 1.11, and 1.11. These figures were greater than the ones regarding direct instruction, which indicate less agreement; however, they were less than the previously discussed item. In its totality, the figures depict minor agreement with this statement although there were diverging responses indicating disagreement with it. These four items in the survey gauged perceptions related to ideas not in line with constructionism. Although there were many responses with all four items that disagreed with the statement, the overarching sentiment was relative agreement. Other than a lack of content knowledge to instruct coding, participants' epistemological and pedagogical underpinnings in these four subgroups pose as an additional challenge. Although these figures do not prevent the integration of constructionism and coding in the classroom, if responses were consistently at the back end of the scale showing disagreement, it would more so buttress the potential to do so.

Other items measured perceptions of statements that were in accord with constructionism's principles. For instance, participants were asked to gauge whether student collaboration and communication are important components of learning. There was significant support for this statement as indicated by means of 1.22, 1.16, 1.28, and 1.30 with a unanimous mode of *strongly agree*. The standard deviations of 0.46, 0.37, 0.56, and 0.82 represented minor departure from the mean. These figures are some of the most robust in the entire set of data and represent participants' belief in the primacy of collaboration and communication, which are crucial for constructionism's and coding's integration in the classroom. Constructionism champions the use of technology in the classroom to allow students to code and create artefacts; thus, the survey prompted a response to whether technology should be a focal tool. The elementary subgroup's responses generated a mean of 2.11, the middle school participants' replies created a mean of 2.34, the high school data produced a mean of 2.51, and the multiple grade cluster's responses yielded a figure of 2.48. The modes for all the respective groups were *somewhat agree* and the standard deviations were 0.99, 1.10, 1.05, and 1.28. There is moderate agreement with technology's primal role in the classroom, but the standard deviation does demonstrate this opinion is not unanimous. This provides some additional support for the integration of coding and constructionism in the classroom, but there may need to be a more convincing commitment to technology. The basis for these responses would need to be investigated as well to shift perceptions toward greater use of technology.

The survey furthermore enquired into whether the focus of teaching is to help students construct knowledge from their learning experiences instead of imparting knowledge. The means were 1.69, 1.61, 1.85, and 1.63, while the modes were *somewhat agree* except for the middle school's predominant selection of *strongly agree*. The standard deviations were 0.72, 0.68, 0.87, 0.88. Although the standard deviations signify the presence of some responses that diverged from the mean, the overall results demonstrate relatively strong agreement with this notion, which offers promise to incorporate constructionism.

The final question analyzed for this set of subgroups directly asked whether constructionism is possible to incorporate into the classroom. Participants initially read a brief explanation of it and proceeded to respond thereby generating means of 1.67, 1.42, 1.62, and 1.26. The scale of scores for this question differed as it ranged from 1–3 and it represented the following choices: easily possible, possible by difficult to do, and not possible. The elementary and high school groups' predominant response was possible but difficult to do, while the middle school and multiple cluster groups primarily selected easily possible. The standard deviations for the subgroups were 0.51, 0.50, 0.54, and 0.45. These figures reveal the agreement that although it is challenging to do so, it is possible to implement constructionism. Among all questions evaluating the potential to integrate constructionism in the classroom, the direct nature of this question coupled with its associated data give credence to this assertion.

Analysis of Teachers in Different School Levels and Administrators. The analysis of the final set of subgroups attempted to evaluate their pedagogical and epistemological positions to determine whether it supports or diminishes the possibility of integrating constructionism and coding in the classroom. The items are listed in Table 35.

Table 35

Descriptive Statistics for Research Question 3

Survey Item	Survey #	*M*	*SD*	*n*
Student collaboration and communication are important components of learning.	55	1.23	0.52	186
Learning means remembering what the teacher has taught.	56	2.89	1.23	186
Teaching is presenting or explaining the subject matter.	57	3.48	1.15	186
Technology should be a focal tool in the classroom.	58	3.66	1.06	186
The focus of teaching is to help students construct knowledge from their learning experience instead of imparting knowledge.	59	1.68	0.79	186
Students should have greater control over their learning process.	60	4.14	0.79	186
Constructionism is possible to apply in the classroom.	61	2.05	1.04	186

Question 61's original scale of 1–3 was recoded to correspond to the same scale for all the other items in the survey because the original purpose was to create a construct. Response Option 1 was easily possible, Option 2 was possible but difficult to do, and Option 3 was not possible. Thus, *easily possible* (1) was adjusted to *strongly agree* (1), *possible but difficult to do* (2) was replaced by *neither agree nor disagree* (3), and *not possible* (3) was recoded to *strongly disagree* (5). Adjustment of the first and third choices was straightforward; however, I struggled with the second choice's modification. Yet, because 2 on the 3-point scale and 3 on the 5-point scale represented the midpoints, and because *possible, but difficult to do* could be interpreted as not overly enthusiastic toward either end of the scale, it was reasonable to adjust it in this manner.

A construct was not created for this set of items due to the low Cronbach's alpha of .45. In assessing the reason for this low value, the direction or the nature of the items were noted as potentially problematic. For instance, direct instruction is regarded as being negative, which was the basis for Question 54; however, student collaboration and communication are noted as being conducive to student learning. Upon my chair professor's review of all the items, it was determined Questions 54, 56, 57, 58, and 60, which were deemed "negative," would be recoded to correspond to the direction of the "positive" questions' responses. Thus, keeping in mind the 1–5 scale of choices, the following recoding system was used: 1 = 5, 2 = 4, 3 = 3, 4 = 2, and 5 = 1. Subsequently, the Cronbach's alpha was recalculated, but the alpha value decreased to .41. There were some item-total correlation values that were negative, such as -0.169, thus those items were deleted. Question 60 was the first item to be removed, which caused the alpha value to increase to .50. However, this remained insufficient to create construct. With these results, question 58 was deleted because its item-total correlation value was .13, which enhanced the alpha to .51. With the current data, Question 61 had the lowest item-total correlation value; however, this question prompted participants to respond to whether constructionism is possible to integrate in the classroom. Therefore, I preferred to not remove the item. One final effort was made retreating a step and reincluding Question 58 and removing Question 59, which had a corrected item-total value of .10. This produced a Cronbach's alpha of .51, which still did not come close to the requisite threshold.

Furthermore, as aforementioned, after I removed 11 duplicate participants from the data set, I reconducted the reliability test like I did for all the other constructs. However, Cronbach's alpha was 0.41, and because the items were not connected sufficiently, a construct was not created. Consequently, instead of offering inferential analysis on these items, they were addressed solely with descriptive statistics.

The descriptive data should be regarded with caution due to the inability to create a construct. Nevertheless, there were some interesting points of data. Questions 55, 59, and 61 focused on teaching strategies that are different than the teacher-centered traditional forms of pedagogy. For instance, Question 55 probed participants to assess whether collaboration and communication are important components of learning. It received an extremely robust mean of 1.23 with a standard deviation of 0.52, which indicates a firm convergence of scores around the mean. The item on teaching's focus to construct student knowledge through learning experiences also received significant support as suggested by the mean of 1.68. The standard deviation was higher compared to Question 55 indicating a more diverse set of responses, but the figures suggest espousal for this statement. The question on whether constructionism is possible to apply also received a noteworthy level of support as suggested by a mean of 2.05, but its standard deviation was greater than 1. Nevertheless, it establishes the specter of using constructionist pedagogies in the classroom, which bodes well for integrating coding instruction.

The other items gauged perceptions about pedagogies and epistemologies that are more in line with a teacher-centered environment. For instance, respondents were asked about the utility of direct instruction, whether learning means what the teacher has taught, and if teaching is presenting or explaining the subject matter. These three items, which corresponded to numbers 54, 56, and 57, had significantly higher means compared to the student-centered questions. This demonstrated participants were not necessarily in favor of those strategies, which enhances the potential to integrate constructionism and coding in the classroom. However, Question 58's mean does pose a challenge. Participants were not in favor of the notion that technology should be a focal tool in the classroom as suggested by a mean of 3.66, yet constructionist philosophy champions its use in the classroom. These beliefs conflict, and additional research is necessary to identify the reasons for this stance on technology. Furthermore, participants were not in favor of students possessing more control in the classroom. Although there was some divergence away from the mean with a standard deviation of 0.79, the mean of 4.14 demonstrably illustrated this viewpoint. Constructionism, on the other hand, deems enhanced student control as critical to learning. Thus, in totality, participants' pedagogical and epistemological beliefs are in consonance with certain constructionist foundations, yet they do somewhat conflict regarding others. Additional research would be necessary to better determine constructionism's feasibility in the classroom.

The subgroup means, which are depicted in Table 36, should be interpreted with caution because the items did not combine to create a construct, yet they are important to report. The means of all five subgroups were remarkably similar, ranging from 2.08 to 2.27. The relatively low standard deviations ranging from 0.4 to 0.52 also illustrate scores clustered close to the mean. Furthermore, unlike the data in the four constructs, which showed certain distinguishing patterns between the subgroups, although inferential analysis was not conducted, no significant points of distinction were present here based on the mean figures. Especially because the values for five items were recoded so responses could be interpreted accurately and uniformly, there can be a relatively firm argument that participants' pedagogical and epistemological beliefs are to a fair degree in line with constructionist tenets because the means indicate they *somewhat agree* with its application. By extension, this enhances the prospects of implementing coding instruction in the classroom.

Table 36

Descriptive Statistics on Subgroups

Subgroup	*M*	*SD*	*n*
Elementary	2.1	0.45	54
Middle	2.08	0.44	35
High	2.27	0.41	34
Multiple clusters	2.06	0.52	25
Administrators	2.19	0.4	38
Total	2.14	0.44	186

Implementation of Coding

The following section will discuss how respondents assessed potential challenges related to implementation. The analysis was multi-tiered; an assessment was conducted on the entire set of data and the data was additionally evaluated based on the position of the respondents, their gender, and grade level of instruction.

Analysis of Total Sample Population. In addition to discussing elements such as the benefits of coding, the literature on coding emphasized means of implementation and its consequent challenges. To gain a more complete understanding of the respondents' perceptions regarding coding, the survey included a block focused solely on this theme. It is included in this portion of the chapter because it enhances the value to the research question on pedagogy and epistemology. Even if educators are amenable to constructionism and prefer to integrate coding in the classroom, implementation challenges abound and must be examined to determine the needs of a school site in effectuating this goal. Table 37 highlights this theme's central measures of tendency.

Table 37

Descriptive Statistics on Implementation Items

Survey Items	*M*	*SD*	Mode
Coding instruction should be required.	2.44	1.06	2
Coding should be a standalone subject.	2.54	1.07	3
Coding should be integrated with other subjects like math, science, and language.	2.14	1.02	2
I can create coding activities at the appropriate level for my students.	3.84	1.41	5
I can integrate coding into my content curriculum.	3.55	1.42	5
A structural change is needed to incorporate this new literacy.	2.05	0.97	2

The last question in the pedagogy and epistemology block asked respondents to state whether they believe constructionism is a feasible framework to apply in the classroom. The scale for this item included the following choices: *easily possible* (1), *possible but difficult to do* (2), and *not possible* (3). The mean of 1.52 and a mode of *somewhat agree* (2) suggested respondents were potentially inclined to do so. Furthermore, the benefits section of the survey indicated strong agreement with coding's abilities to prepare students for various occupations in the future, diversify representation in STEM occupations, and foster numerous skills including collaboration, problem-solving, and logical thinking. However, when respondents contemplated implementation issues, the results were somewhat dissimilar and not necessarily in accordance with the aforementioned considerations regarding coding.

When prompted to opine on whether coding instruction should be required, the mean was a 2.44, the standard deviation was 1.06, and the mode was *somewhat agree* (2). In totality, although there was some agreement, the data indicate participants were not necessarily in favor of requiring coding instruction in the curriculum. Those who strongly or *somewhat agreed* with this statement were then requested to respond to two additional queries focused on the grade levels that should be exposed to coding instruction and the length of weekly time dedicated to it, which were evaluated with a frequency measure. All grade levels ranging from kindergarten to 12th grade were chosen with a relatively equitable distribution; however, more participants chose Grades 3–12. Additionally, most of these participants chose the 60-minute mark as the preferred length of time dedicated to coding instruction on a weekly basis as opposed to 15, 30, or more than 60 minutes.

Although many participants strongly or *somewhat agreed* with coding's requirement in the curriculum, the mean and standard deviations also suggest some participants were not completely in favor of this notion. This may be attributed to the inherent challenges to integrate its instruction in the classroom stemming from the perspective of the educator, the student, and the administration. In regard to challenges for the teacher, the survey offered choices including insufficient time, lack of knowledge and preparation to instruct it, lack of access to technology, and insufficient student interest. Participants were allowed to choose more than one answer choice and 119 participants chose inadequate time, 172 selected educator lack of preparation, 110 emphasized a deficiency in available technology, and 42 respondents believed student interest was a hindrance. Although there are various impediments to implementation, lack of teacher preparation was the response choice chosen with the greatest frequency. Participants were also given the opportunity to state additional challenges not offered as fixed choices. Quite a few responses indicated a lack of teacher interest and passion.

The overarching assertion from this question parallels the responses on two additional inquiries, which were evaluated with a frequency measure, asking participants to state their competency level in teaching block- and syntax-based coding. These questions were scored with a different scale ranging from 1 = *highly capable* to 4 = *not capable*. Nineteen stated they were highly capable in teaching block-based coding using applications such as Scratch, 26 stated they were capable, 22 perceived themselves as somewhat capable, but 121 averred they were not capable. Participants considered themselves to be even less capable in teaching syntax-based coding such as Python. Nine posited there were highly capable, 15 were capable, three chose somewhat capable, and 140 deemed themselves to not be capable. These data once again confirm the lack of preparation and training among teachers to either impart coding lessons separately or integrate it in the instruction of other courses.

The question on challenges was posed once again, but from a student's perspective with fixed response choices, including lack of prior student experience with coding, deficient interest, lack of access to technology, and perception that it is difficult. Educators suggested prior experience with coding and the perception of difficulty were the primary inhibiting factors for students. Other responses included student lack of confidence in math or science, which are subjects commonly associated with coding. This question was also assessed from the administration's perspective with response choices including lack of finances to purchase requisite technology, perception of inadequate importance to warrant utility of resources, and lack of educator preparation. An option for additional responses was also given. Lack of technology and insufficient teacher preparation were chosen with the most frequency. This is in line with teacher perceptions about their own lack of preparation to impart coding instruction. In regard to challenges not offered as response choices, participants identified limited time and scheduling issues, a concern that students emboldened by coding skills will use it for malicious purposes, and lack of resources to employ a coding teacher. Insufficient time and the impact on the regular schedule were concerns voiced by teachers as well. Time is a precious commodity, and the daily curriculum is already sated; therefore, it is understanding that an additional component would cause consternation among educators.

Furthermore, the statement about coding serving as a separate subject yielded a mean of 2.54, a standard deviation of 1.07, and a mode of *neither agree nor disagree* (3). There was more agreement with the idea that coding should be integrated with subjects such as math, reading, and science as demonstrated by a mean of 2.14, a standard deviation of 1.02, and a mode of *somewhat agree* (2). However, the sentiment here is not completely compatible with the notion that coding should be required. It is possible that although educators are amenable with the precepts of constructionism and to integrating coding instruction, they do not necessarily believe it should be compulsory. This notion can be attributed to the reality that most educators who participated in the survey were employed in schools that did not require coding. The means for the two questions on whether participants can create and integrate coding activities into the curriculum were higher reaching close to 4 with a mode of *strongly disagree* (5), which lucidly indicates their lack of preparation and/or confidence to take this step in the classroom. Nevertheless, participants agreed with the statement that a structural change is necessary to integrate coding into the curriculum. The mode was *somewhat agree* (2), the mean was 2.05, and the standard deviation was 0.97. Once again, the measures for this item seem to be inconsistent with the ones from the statement probing into whether coding instruction should be required. Although participants were not necessarily completely in support of requiring coding instruction, they were more so in agreement with the need to impose a structural change to incorporate it. It is possible that participants believe a structural alteration will lend itself to preparing educators to integrate coding instruction.

Analysis of Data Based on Position. The implementation of coding is a crucial component of this study. Consequently, data for subgroups based on position or role was analyzed, which is detailed in Table 38. The three primary groups of teachers, administrators, and participants with multiple roles were queried regarding the challenges teachers may experience with coding instruction. This item attempted to appraise teacher challenges from the perspective of administrators to evaluate whether there is any parallel between the subgroups' responses. Specific response choices for this item included insufficient time (1), lack of knowledge/preparation to teach (2), lack of access to technology (3), lack of student interest (4), and an option for an alternative reason (5). Participants were allowed to choose multiple answers.

Table 38

Frequency for Implementation Items (Based on Positions)

Item	Response Choices	Teacher			Admin			MR		
		F	%	*n*	*F*	%	*n*	*F*	%	*n*
What challenges do you think teachers may face with coding instruction in the classroom (you may choose more than one answer choice)?	Insufficient time	71	27.95	254	21	23.86	88	24	24	100
	Lack of knowledge/preparation to teach	95	37	254	37	42.04	88	37	37	100
	Lack of access to technology (i.e., devices, wi-fi network)	60	24	254	20	22.72	88	26	26	100
	Lack of student interest	24	9.45	254	7	7.95	88	11	11	100
	Other (please specify)	4	1.57	254	3	3.40	88	2	2	100
What challenges do you think administration may encounter in offering coding instruction in the classroom (you may choose more than one answer choice)?	Lack of resources to purchase technology (i.e., student devices, enhanced wi-fi)	77	35.64	216	24	35.29	88	30	36.58	82
	Do not perceive it as being important enough to warrant time and money	44	20.37	216	6	8.82	88	16	19.51	82
	Educators have not been prepared to teach it in the classroom	91	42.13	216	36	40.91	88	34	41.46	82
	Other (please specify)	4	1.85	216	2	2.94	88	2	2.43	82
What grade levels do you think coding should be instructed in?	K–2nd	21	13.54	155	9	17.64	51	16	19.75	81
	3rd–5th	40	25.81	155	13	25.49	51	19	23.45	81
	6th–8th	48	30.97	155	17	33.33	51	23	28.39	81
	9th–12th	46	29.68	155	12	23.52	51	23	28.39	81
How should students learn coding in school (you may choose more than one answer choice)?	Teacher driven followed by individual student work	39	26.71	146	13	28.89	45	17	28.33	60

Table 38 (continued)

Frequency for Implementation Items (Based on Positions)

Item	Response choices	Teacher			Admin			MR		
		F	%	*n*	*F*	%	*n*	*F*	%	*n*
How should students learn coding in school (you may choose more than one answer choice)?	Some teacher guidance followed by collaborative student work	87	59.59	146	31	68.89	45	37	61.67	60
	Teacher is not involved; only student collaborative work	10	6.85	146	0	0	45	4	6.67	60
	Other (please specify)	10	6.85	146	1	2.22	45	2	3.33	60
How effectively can you teach block-based coding programs like Scratch?	Highly capable	10	9.26	108	1	2.63	38	7	18.42	38
	Capable	12	11.11	108	4	10.52	38	9	23.68	38
	Somewhat capable	14	12.96	108	5	13.15	38	3	7.89	38
	Not capable	72	66.67	108	28	73.68	38	19	50	38
How effectively can you teach syntax-based coding programs such as Python?	Highly capable	6	5.71	105	0	0	38	3	7.89	38
	Capable	5	4.76	105	4	10.52	38	5	13.15	38
	Somewhat capable	12	11.42	105	4	10.52	38	7	18.42	38
	Not capable	85	80.95	105	30	78.95	38	23	60.52	38

Note. MR = multiple role.

The frequency for teacher responses was 254; the predominant responses were insufficient time (27.95%), lack of knowledge and preparation to instruct it (37%), and lack of access to technology (24%). The sample size for administrator responses was 88 and the first three response choices were the primary ones selected. 23.86% of the responses were insufficient time, 42.04% of the responses were lack of knowledge/preparation, and 22.72% of the responses were lack of access to technology. The total frequency of responses among participants with multiple roles was 100. Similarly to the teachers' and administrators' chief selections and the resulting percentages, the multiple roles subgroup principally identified insufficient time (24%), lack of knowledge and preparation to instruct it (37%), and lack of access to technology (26%)

as the most pressing impediments to coding instruction in the classroom. Furthermore, 11% of the responses identified lack of student interest as a challenge as well. Across all three subgroups, lack of preparation and knowledge to instruct coding was perceived to be the greatest hindrance to coding instruction in the classroom. This sentiment is in consonance with the prevailing literature describing teacher perceptions. To integrate coding instruction effectively and efficiently, teachers require the appropriate pedagogical and content foundations.

A similar question was asked, but instead it focused upon administrator challenges with coding instruction. Similarly to administrators' perceptions regarding teacher challenges explained above, it was also significant to assess administrator challenges from the perspective of teachers. To create a system effectively, whereby coding instruction is implemented in the classroom, it is important that these different stakeholders understand the primary challenge/s to effecting it. This will better position a school to address these obstacles and overcome them. The response choices were lack of resources to purchase the requisite technology (1), do not perceive it as being important enough to warrant time and money (2), educators have not been prepared to teach it in the classroom (3), and an option to choose an alternative reason (4). Participants were allowed to select manifold choices. The frequency for teachers was 216 and lack of resources comprised 35.64% of this total number, the second option was selected 20.37% of the time, and lack of educator preparation was the primary response choice with 42.13%. The administrators' responses were quite similar in that 35.29% of the total frequency of 88 went to option 1 and 40.91% of the responses cited lack of educator preparation as the chief obstacle. The responses of the final subgroup also mimicked those from the first two subgroups. The chief response choice selected was lack of teacher preparation, which amounted to 41.46% of the 82 total number of responses chosen, whereas lack of resources amounted to 36.58% of the selections. Not perceiving it important enough to justify allocation of time and money came in at about 19.51%. The overarching sentiment from this item's data is in harmony with the literature. Although there are multifaceted challenges to integrating coding instruction in the classroom, lack of proper preparation of teachers is a significant obstacle requiring attention by each school and district/diocese.

Many countries have instituted coding beginning in the primary/elementary levels. This survey gathered data on perceptions surrounding the grade levels that coding should be instructed in; however, this question was only prompted if participants strongly or somewhat believed coding instruction should be required in a preceding question. There were 55 teachers, 18 administrators, and 26 individuals with multiple roles who answered this question. The response choices were *K–second grade* (1), *third–fifth grades* (2), *sixth–eighth grades* (3), and *ninth–twelfth grades* (4) and participants were allowed to choose more than one response. The total number of choices selected by teachers was 155 and there was a relatively equitable spread between third–twelfth grades. The K–second cluster received 13.54% of the total tally, but the subsequent choices received 25.81%, 30.97%, and 29.68% of the total. This indicates their belief that coding instruction is more appropriate beyond the K–second level. The number of choices chosen by administrators was 51 and 17.64% of this number was selected for K–second, third–fifth received 25.49% of the tally, 33.33% was allocated to sixth–eighth, and high school received 23.52% of the vote. The largest differences between administrators and teachers was that more of the former group believed coding instruction would be appropriate for the K–second level while less perceived it to be important at the high school level.

The final subgroups' data more was analogous to the teachers' responses. 26 individuals chose 81 responses and among this total, 19.75% identified K–second, 23.45% selected third–fifth, and 28.39% of the selections elected middle and high schools. Overall, there seems to be support for coding instruction in all grade levels, but slightly less for the K–second cluster.

Another question to ascertain the possibility of implementation queried participants' perceptions on how students should learn coding. The offered choices were the following: *teacher driven followed by individual student work* (1), *some teacher guidance followed by collaborative student work* (2), *teacher is not involved; only student collaborative work* (3), and an option for an alternative text response, notated by *other* (4). Participants were allowed to choose more than one response. The tally for teachers was 146 and 26.71% of these responses chose option one, while almost 60% of the responses chose option two. Administrators' responses were even more pronounced toward the first two choices as 28.89% selected the first option and 68.89% preferred the second one. The final subgroup's data were in consonance with these data as well because over 28% of the 60 responses selected option one, while 61.67% picked Option 2. In totality, the preference among all subgroups was a pedagogical strategy that first placed the teacher at the fore of instruction, but it then offered students the opportunity to collaboratively work to apply what was learned.

An additional set of questions evaluated the level of coding knowledge among the three subgroups. The first question asked if the participant can teach block-based coding programs like Scratch, while the next question asked if they can effectively instruct syntax-based coding like Python. Both questions offered the following response choices: *highly capable* (1), *capable* (2), *somewhat capable* (3), *not capable* (4). The teacher subgroup's predominant response choice was not capable at almost 67%, the administrator subgroup's primary selection was not capable at nearly 74%, and the final subgroups foremost election was not capable as well reaching 50%. However, 18.42% of this group's responses were highly capable and almost 24% of it was capable. The reason for this difference relative to the other subgroups is that it contains people who hold a STEM position or are associated with STEM at their respective sites. Consequently, they would be more comfortable teaching at least relatively simpler coding programs through Scratch or other similar apps or websites.

Regarding efficacy in teaching syntax-based programs, all three groups principally chose not capable with percentages of nearing 81%, 79%, and 60.5%. Once again, the last group, which contains people with a STEM background, perceived themselves as more proficient at teaching syntax-based coding. This form of coding is more complex and thus, about 10% more of the respondents in the final group stated they were not capable relative to their responses for block-based coding. Such high numbers of perceived inefficacy are not only concordant with the literature, but it unequivocally underscores the need to invest resources into properly training teachers and administrators alike to integrate coding in school.

Additional questions attempted to gauge participants' preparedness of incorporating coding instruction in the classroom. These included whether participants can create appropriate coding activities based on the grade level of instruction and whether they can integrate it in the content curriculum. The means for the subgroups were 3.99, 4.02, and 3.23, and the modes were all *strongly disagree*. The standard deviations were 1.32, 1.28, and 1.58. These data continue to highlight the need for teacher preparation, as it is evident all three subgroups do not consider themselves adept at creating coding activities. Although the mean of 3.23 for the last subgroup was considerably lower than the other subgroups, the large standard deviation of 1.58 and the mode of 5 nevertheless signifies a large number of respondents do not perceive themselves as capable. The question regarding integration of coding in the content curriculum once again revealed perceived lack of effectiveness. The means of 3.82, 3.55, and 2.73 were somewhat lower across all three groups, the modes of 5 for the first two subgroups remained the same, and the standard deviations of 1.28, 1.34, 1.57 were consistent with the figures in the previous question. However, the anomaly in the data was the mode of *strongly agree* for the multiple role subgroup. Although the mean was closer to *neither agree nor disagree*, the large standard deviation here signified there are a considerable number of individuals in this subgroup who can integrate coding in the classroom; however, there remained those who found themselves on the other end of the spectrum.

The last piece of evaluated evidence stemmed from a question on whether a structural change in curriculum is needed to incorporate coding. The means for the subgroups were 2.01, 2.34, and 1.81, and all the modes were *somewhat agree*. Standard deviations of 1.00, 1.02, and 0.75 showed the existence of some responses that did not gather around the mean. Nevertheless, these data revealed some agreement toward a need to effect structural change in the curriculum to incorporate coding. This may take the form of adding a coding class into the curriculum or increasing the length of science, math, social studies, and language courses to provide additional time to the teacher to integrate coding into the content. Regardless of the ultimate route, the current iteration of the curriculum is not necessarily equipped to integrate coding's instruction.

Analysis of Data Based on Gender. This study also analyzed implementation questions based on the gendered subgroups of female and male (see Table 39). For instance, participants were asked about the challenges teachers may face with coding instruction. Response choices for this item included insufficient time (1), lack of knowledge/preparation to teach (2), lack of access to technology (3), lack of student interest (4), and an option for an alternative reason (5). Participants were allowed to choose multiple answers; thus, the number of female responses was 339.

Table 39

Frequency for Implementation Items (Based on Gender)

		Female			Male		
Item	Response Choices	*F*	%	*n*	*F*	%	*n*
What challenges do you think teachers may face with coding instruction in the classroom (you may choose more than one answer choice)?	Insufficient time	95	28.02	339	22	21.15	104
	Lack of knowledge/preparation to teach	130	38.35	339	41	39.42	104
	Lack of access to technology (i.e., devices, wi-fi network)	84	24.78	339	23	22.11	104
	Lack of student interest	25	7.37	339	16	15.38	104
	Other (please specify)	5	1.47	339	2	1.92	104
What challenges do you think administration may encounter in offering coding instruction in the classroom (you may choose more than one answer choice)?	Lack of resources to purchase technology (i.e., student devices, enhanced wi-fi)	99	35.23	281	29	34.52	84
	Do not perceive it as being important enough to warrant time and money	52	18.50	281	16	19.05	84
	Educators have not been prepared to teach it in the classroom	124	44.12	281	37	44.04	84

Table 39 (continued)

Frequency for Implementation Items (Based on Gender)

Item	Response Choices	Female			Male		
		F	%	*n*	*F*	%	*n*
What challenges do you think administration may encounter in offering coding instruction in the classroom (you may choose more than one answer choice)?	Other (please specify)	6	2.13	281	2	2.38	84
What grade levels do you think coding should be instructed in?	K–2nd	38	16.45	231	8	13.11	61
	3rd–5th	59	25.54	231	15	24.59	61
	6th–8th	70	30.30	231	20	32.78	61
	9th–12th	64	27.70	231	18	29.50	61
How should students learn coding in school (you may choose more than one answer choice)?	Teacher driven followed by individual student work	46	24.73	186	23	35.50	63
	Some teacher guidance followed by collaborative student work	119	63.98	186	35	55.55	63
	Teacher is not involved; only student collaborative work	11	5.91	186	2	3.17	63
	Other (please specify)	10	5.37	186	3	4.76	63
How effectively can you teach block-based coding programs like Scratch?	Highly capable	11	7.80	141	8	18.60	43
	Capable	18	12.76	141	8	18.60	43
	Somewhat capable	17	12.05	141	4	9.30	43
	Not capable	95	67.37	141	23	53.59	43
How effectively can you teach syntax-based coding programs such as Python?	Highly capable	4	2.84	141	5	11.62	43
	Capable	7	4.96	141	8	18.60	43
	Somewhat capable	19	13.47	141	5	11.62	43
	Not capable	111	78.72	141	25	58.13	43

The primary responses chosen were insufficient time which garnered 28.02% of the tally, lack of teacher knowledge and preparation gathered 38.35% of the total number, and lack of access to technology was chosen 24.78% of the time. The male responses, which equaled 104, mimicked the females' selections. Lack of teacher knowledge and preparation was the choice selected with the greatest frequency. This cited challenge is not only consonant with the prevailing literature, but so too with the selection of the subgroups of teachers, administrators, and participants with multiple roles.

The same question was posed from an administrator's perspective. The response choices were lack of resources to purchase the requisite technology (1), do not perceive coding as being sufficiently important to warrant time and money (2), educators have not been prepared to teach it in the classroom (3), and an option to choose an alternative reason (4). Participants were allowed to select multiple choices. Identical to the chief challenge of teachers as selected by male and female participants, the primary response choice for the greatest administrator challenge was lack of teacher preparation and knowledge as well. Among the tally of 281 female responses, that option was chosen 44.12% of the time, and males selected it with an incidence of 44.04%.

Participants were also queried about what grade levels should coding be instructed in. This question was only prompted if participants strongly or somewhat believed coding instruction should be required in a preceding question. There were 231 female and 61 male responses, which was due to the ability choose more than one response. The choices were *K–second* (1), *third–fifth* (2), *sixth–eighth* (3), and *ninth–twelfth* (4). There was a fair female distribution across all four choices, as indicated by 16.45%, 25.54%, 30.30%, and 27.70%. Male participants' responses were similar with 13.11%, 24.59%, 32.78%, and 29.50%. This demonstrates both genders deem coding to be important for all grade levels but most especially in the elementary, middle, and high school levels.

The analysis of implementation continued with straightforward questions seeking to understand how effectively participants can instruct coding. Two separate items, with one focusing on efficacy of teaching block-based coding, and another honing on effectiveness of instructing syntax coding, were posed. Both questions offered the response choices of *highly capable* (1), *capable* (2), *somewhat capable* (3), and *not capable* (4). Female participants primarily chose “not capable” to explain their level of efficacy at teaching both types of programming. Although some participants chose other responses, the percentages of 67.37 and 78.72 represented the high degree of unpreparedness to instruct either form of coding. The male responses, however, generated a wider distribution across all four choices as indicated by 18.60%, 18.60%, 9.30%, and 53.59%. There is a healthy swath of male participants who deemed themselves as highly and somewhat capable at instructing block-based coding; however, like the female output, most male respondents selected not capable. Regarding syntax-based coding, the percentages were relatively similar with not capable being chosen with the greatest frequency and thus holding 58.13% of responses. These data points once more accentuate the need to prepare teachers to instruct coding in the classroom.

The final set of questions on the potential to implement coding instruction in the classroom urged participants to perceive their ability to create coding activities at the appropriate level in the classroom and whether they can integrate it into their content curriculum. Female participants’ responses yielded means of 3.94 and 3.6 with a mode of *strongly disagree*. The standard deviations were 1.37 and 1.42, respectively, which signaled some varied responses away from the mean, including those on the front end of the Likert scale. Nevertheless, the relatively high means and modes of *strongly disagree* lend further credence to the notion that educators need resources, support, and preparation to immerse coding instruction in the classroom. Male responses demonstrated a slight difference in perception as typified by means of 3.44 and 3.37, but the modes of *strongly disagree* further the identical point that there is a general need to provide preparatory resources for all teachers. The survey also measured whether participants believe a structural change in curriculum is necessary to incorporate coding. Females responded to this query with a mean of 2.02, a mode of *somewhat agree*, and a standard deviation of 0.95. Males responded with almost identical selection as demonstrated by a mean of 2.04, a mode of *somewhat agree*, and a standard deviation of 0.95. Although there was not necessarily a fervent perception that structural change is necessary, the data does nonetheless exhibit sizable support for this idea.

Analysis of Data Based on Grade Clusters. The first question analyzed for these subgroups (elementary, middle school, high school, and participants who instruct in multiple clusters) was based on the perceptions of challenges teachers may experience with coding instruction. Participants were allowed to choose multiple options: insufficient time (1), lack of knowledge and preparation to teach (2), lack of access to technology (3), lack of student interest (4), and an option for an alternative response (5). The total number of responses for the elementary subgroup equated to 132, the middle school's response frequency was 92, the high school chimed in with 88 responses, and the final group's total responses were 88 as well. Across all four subdivisions, lack of teacher preparation and knowledge received the most responses by a healthy margin; however, insufficient time and lack of access to technology received considerable attention as well. Table 40 provides a summation of the data.

Table 40

Frequency for Implementation Items (Based on Grade Cluster)

Item	Response Choice	Elem			Mid			High			MC		
		F	*%*	*N*	*F*	*%*	*N*	*F*	*%*	*N*	*F*	*%*	*N*
What challenges do you think teachers may face with coding instruction in the classroom (you may choose more than one answer choice)?	Insufficient time	41	31.06	132	26	28.26	92	17	19.32	88	21	23.86	88
	Lack of knowledge or preparation to teach	49	37.12	132	35	38.04	92	36	40.91	88	31	35.23	88
	Lack of access to technology (i.e., devices, wi-fi network)	32	24.24	132	21	22.82	92	21	23.86	88	25	28.41	88
	Lack of student interest	9	6.81	132	9	9.78	92	11	12.50	88	10	11.36	88
	Other (please specify)	1	0.76	132	1	1.08	92	3	3.41	88	1	1.13	88
What challenges do you think administration may encounter in offering coding instruction in the classroom (you may choose more than one answer choice)?	Lack of resources to purchase technology (i.e., student devices, enhanced wi-fi)	38	34.54	110	32	39.02	82	24	33.33	72	20	35.71	56
	Do not perceive it as being important enough to warrant time and money	23	20.91	110	18	21.95	82	13	18.05	72	9	16.07	56
	Educators have not been prepared to teach it in the classroom	49	44.54	110	32	39.02	82	32	44.44	72	24	42.86	56
	Other (please specify)	0	0.00	110	0	0.00	82	3	4.17	72	3	5.36	56

Table 40 (continued)

Frequency for Implementation Items (Based on Grade Cluster)

Item	Response Choice	Elem			Mid			High			MC		
		F	*%*	*n*	*F*	*%*	*n*	*F*	*%*	*n*	*F*	*%*	*n*
What grade levels do you think coding should be instructed in?	K–2nd	13	15.29	85	7	14.28	49	7	13.20	53	15	19.74	76
	3rd–5th	22	25.88	85	13	26.53	49	10	18.86	53	21	27.63	76
	6th–8th	27	31.76	85	16	32.65	49	16	30.19	53	21	27.63	76
	9th–12th	23	27.05	85	13	26.53	49	20	37.73	53	19	25	76
How should students learn coding in school (you may choose more than one answer choice)?	Teacher driven followed by individual student work	18	25.00	72	11	20	55	19	33.33	57	16	33.33	48
	Some teacher guidance followed by collaborative student work	44	61.11	72	34	61.82	55	34	59.65	57	29	60.42	48
	Teacher is not involved; only student collaborative work	6	8.33	72	6	10.90	55	1	1.75	57	1	2.08	48
	Other (please specify)	4	5.55	72	4	7.27	55	3	5.26	57	2	4.17	48
How effective-ly can you teach block-based coding programs like Scratch?	Highly capable	2	3.63	55	3	7.89	38	4	10.25	39	9	27.27	33
	Capable	8	14.55	55	7	18.42	38	3	7.69	39	4	12.12	33
	Somewhat capable	5	9.09	55	6	15.79	38	4	10.25	39	4	12.12	33
	Not capable	40	72.72	55	22	57.89	38	28	71.79	39	16	48.48	33

Table 40 (continued)

Frequency for Implementation Items (Based on Grade Cluster)

Item	Response Choice	Elem			Mid			High			MC		
		F	%	*n*	*F*	%	*n*	*F*	%	*n*	*F*	%	*n*
How effectively can you teach syntax-based coding programs such as Python?	Highly capable	1	1.82	55	1	2.63	38	4	10.25	39	3	9.09	33
	Capable	3	5.45	55	1	2.63	38	1	2.56	39	6	18.18	33
	Somewhat Capable	5	9.09	55	7	18.42	38	3	7.69	39	6	18.18	33
	Not capable	46	83.64	55	29	76.31	38	31	79.49	39	18	54.55	33

Note. MC = multiple cluster.

The same question was posed, but in consideration to administrator challenges. The response choices were *1ack of resources to purchase technology such as student devices* (1), perception that *it is not important enough to warrant time and money* (2), *lack of teacher preparation to instruct it in the classroom* (3), and an option for an alternative response, indicated by *other* (4). Across all four subgroups, lack of teacher preparation was the prevalent response choice with 44.54%, 39.02%, 44.44%, and 42.86%, but lack of resources to purchase technology was not too far behind.

Another critical question to assess the prospect for implementation was the grade levels in which coding should be instructed. This question was only prompted if participants strongly or somewhat believed coding instruction should be required in a preceding question. Participants were allowed to choose more than one answer choice among K–second (1), third–fifth (2), sixth–eighth (3), and ninth–twelfth (4). There was a relatively equitable distribution between all the grade clusters, but middle school was the most favored choice as indicated by the greatest percentage among three out of the four subgroups. The high school subdivision chose its own grade cluster as the most suitable cluster in which to instruct coding.

Participants were also probed about their ability to teach block-and-syntax based coding. The offered choices were *highly capable* (1), *capable* (2), *somewhat capable* (3), and *not capable* (4). Regarding block-based coding, 72.72% of the elementary group stated they were *not capable*, the middle school respondents responded in kind with 57.89%, 71.79% of high school respondents perceived themselves as *not capable*, and 48.48% of participants in the multiple cluster groups stated the same. Of note, 27.27% of the last group did suggest they were *highly capable* at instructing block-based coding. As for syntax-based coding, the data were even more pronounced, indicating an extremely significant swath of participants incapable of instructing it. The percentages of 83.64%, 76.31%, 79.49%, and 54.55% evince this point. Eighteen percent of respondents in the multiple cluster group believed they were *somewhat capable*; however, such stark figures corroborate the literature and preceding evidence from other subgroups highlighting the imperative need to develop educators' capacity to instruct coding in the classroom.

Supplementary questions appraising participants' perceptions of self-efficacy focused on their ability to create coding activities at the appropriate level for students, their capacity to integrate coding into the content curriculum, and whether a structural change is required to incorporate its instruction. The responses of all four subgroups for the first question illustrated their unease. The means of 4.06, 3.60, 4.07, and 3.07, coupled with a unanimous mode of *strongly disagree,* highlight unpreparedness. The final subgroup, composed of some CS or technology teachers, demonstrated a relative sense of preparation because their mean was in the middle of the scale. This is not unexpected because this group has teachers who are comfortable with coding instruction and thus consistently demonstrated they are most prepared to impart it as indicated by their relative understanding of what coding entails, their positive perception regarding its benefits, and their like mindedness with core constructionism tenets. However, the standard deviations of 1.27, 1.38, 1.31, and 1.71 were wide ranging and significant, which suggest a healthy divergence away from the mean for all four subdivisions. The standard deviations, especially for the final group, demonstrated there were respondents who were more adept and less comfortable with creating coding activities, yet the data principally show participants were generally not capable. This is another piece of data underscoring the need for teacher preparation to integrate coding in the curriculum.

An additional question directed toward participants' capacity to integrate coding into the content curriculum attempted to gauge whether they can immerse coding activities in different subjects. The literature explains it is effective and efficient to supplement content with coding activities because students will learn how to code while reviewing key lessons. Like responses regarding the capacity to create coding activities at the appropriate grade level, participants' answers produced means of 3.76, 3.47, 3.87, and 2.59 with modes of *strongly disagree*, except for the final group's chief response of *strongly agree*. The mean of this subgroup was the lowest and by a good margin; although the data consistently exhibited their positive attitude toward coding, their indices for this question were unanticipated and inconsistent with the previous question. This subgroup's mode for the previous question was *strongly disagree*, yet it completely reversed when prompted to respond to their ability to immerse coding into the content curriculum. These two questions are not dissimilar; therefore, the significant discrepancy would need to be further investigated to accurately describe it. Aside from the final subgroup's perceptions, the other three subdivisions once again noted unease with integrating coding in the classroom.

The final question assessed for the third research question from the vantage point of these four subgroups urged respondents to consider whether a structural change is necessary to immerse coding into the curriculum. The means were 1.98, 2.26, 1.92, and 1.89 with modes of *somewhat agree* for the first three groups, while the last subgroup *strongly agreed* with this statement. The standard deviations were 0.90, 1.22, 0.70, and 1.05, which exhibit some variance away from the mean. Once again, the final subgroup demonstrated the most fervor toward this item, indicating their perception that coding is necessary in the classroom. The other three subgroups also affirmed a degree of agreement.

Third Research Question's Conclusion

Pedagogical and epistemological philosophies and implementation challenges were evaluated to determine whether constructionism and coding can be integrated in ADLA Catholic school curriculums. The data evaluating the entire sample population demonstrated respondents did not necessarily consider themselves as the focal point of the classroom and although they asserted there is some utility in direct instruction, student collaborative opportunities, enhanced student control during the learning process, and construction of knowledge from learned experiences were pedagogical underpinnings of the educators. These assertions were evident in the data stemming from teacher, administrator, multiple role participants, males, females, elementary, middle, high school, and multiple cluster subgroups as well. Furthermore, the general population of respondents and all the analyzed subgroups only *somewhat agreed* technology should serve as a principal tool in the classroom. Participants also did not necessarily believe relinquishing control in the classroom to students would be efficacious. Although there was some general belief that constructionism is feasible to apply in the classroom, participants' perceptions, in totality, signify they are not completely prepared to do so at this point. Nevertheless, the data did paint a hopeful picture that this may be possible in the future. For instance, the three subgroups of teachers, administrators, and those with multiple roles perceived a chief tenet of constructionism, which is students' ability to create knowledge from their learned experiences, to be crucial. The data on this question from the different cluster of instruction subdivisions, on the other hand, showed more of a modest enthusiasm. However, these respondents deemed some utility in direct instruction, but considered student collaboration and communication especially vital to learning. Moreover, their firm assertion that constructionism is possible to apply in the classroom lends hope to realizing this possibility. Moreover, according to teachers, administrators, participants with multiple roles, and female and male respondents, coding was believed to be best suited for the third–twelfth grade levels although there was a smaller percentage of these population groups who believed K–second could benefit as well.

The final analysis, which focused on the five subgroups of elementary, middle, high, multiple cluster, and administration, demonstrated participants' pedagogical and epistemological beliefs are, to a fair degree, not in consonance with constructionist tenets. For instance, they regarded student collaboration and communication as crucial elements of student learning. However, they did not necessarily believe technology should be a focal tool in the classroom or that students should possess greater control in the classroom. Constructionist philosophy suggests collaboration and communication are critical and champion enhanced student control in the classroom and the centrality of technology. These conflicting viewpoints, at least as they relate to constructionism, do pose questions about its feasibility in the classroom, which, in turn, affect the potential integration of coding.

Items focused on implementation of coding highlighted educator concerns including, but not limited to, inadequate resources to purchase necessary technology, insufficient time, lack of student experience, and deficient teacher interest. The greatest impediment, however, was an absence of teacher preparation to impart coding instruction. This general sentiment was corroborated when a preponderance of respondents affirmed they are incapable of instructing block-based or syntax-based coding. Furthermore, additional credence was lent to this point from the data on the subgroups of teachers, administrators, and individuals with multiple roles. Moreover, analysis of female and male subgroups as well as teachers in the different grade clusters revealed lack of teacher preparation as the chief challenge to integrating coding instruction from the perspective of teachers and administrators. The data on male and female respondents also highlighted a firm discomfort with instructing block and syntax-based coding. Female and male teachers were also generally uncomfortable with creating coding activities and integrating it into the content curriculum. Of further note was these two subgroups' relatively strong belief that a structural change in the curriculum is required to merge coding instruction into the curriculum. These sentiments were harbored by teachers in the different grade clusters as well. Other than the final subgroup of teachers in different grade clusters, who showed some capacity to impart coding instruction, all other subgroups intimated a strong discomfort with teaching block-based or syntax-driven coding as well as the need to restructure the curriculum to include its instruction. The items on pedagogy and epistemology signified the possibility of integrating constructionism and coding in the classroom; however, implementation challenges complicate this endeavor. Even if educators believe in the benefits of coding and constructionism and are wholeheartedly willing to integrate its tenets, the lack of ability to do so is a significant hindrance. To implement coding and constructionism, tangible steps must be taken by each school site to overcome the identified impediments.

Summary and Conclusion

The first research question gauged the respondents' understanding of coding. The data demonstrated participants generally understood what coding entails; however, they lacked a more detailed understanding of its nuances. Furthermore, the data emphasized the value of coding and the students' desire to learn this literacy. However, the data indicated there is a disparity in access to technology for different ethnic groups. There was also a belief that the same segments of the population are not equitably represented in STEM occupations. Moreover, the data painted a similar picture regarding females, especially as it relates to STEM occupation representation. However, quite resolutely, respondents asserted that females and males can succeed in coding and that there is no inherent advantage due to gender. The four sets of analyses of the various subgroups highlighted that perception that participants have a relatively firm understanding of what coding generally entails; however, a more nuanced understanding is lacking. In line with this sentiment, participants did not consider themselves adept at instructing block- or syntax-based coding nor able to integrate it with the instruction of content curriculum. The only group deviating from this trend slightly was the multiple role and multiple cluster groups, which contained personnel who taught CS and technology.

The second research question assessed the perceived benefits of coding as it relates to academic achievement, occupational preparedness, and the capacity to diminish the disparity in

STEM representation. The respondents strongly believed in coding's ability to foster skills such as critical thought, problem-solving, creativity, and collaboration skills. Furthermore, the data suggested the belief that coding enhances student achievement in different subjects and better situates students to secure technology and to a smaller degree non-technology related positions. These notions were also evident in the analyses of all the sets of subgroups. Moreover, participants asserted coding instruction can be a conduit for enhanced female and ethnic group interest in K–12 STEM subjects and diversified representation in such occupations.

The final research question probed the potential linkage between educator pedagogy and epistemology and the potential to incorporate constructionism and coding in the classroom. Although participants did consider direct instruction to have utility, the data also signified the primal role of student collaboration and communication along with the need to construct knowledge from their learned experiences. These points of data were additionally noted in the subgroup analyses. Furthermore, the item querying whether technology should possess a focal role in the classroom received a relatively tepid response, although most participants asserted they *somewhat agreed* with the statement. The subgroups as well did not fervently consider technology to be a primal tool in the classroom. Nevertheless, and very importantly, the data indicated the belief that constructionism is feasible to apply in the classroom. This is a powerful piece of data because this perception can become a foundational block toward integrating coding and constructionism in the classroom in the future. Although participants agreed with some of constructionism's tenets, their perceptions conflicted with others, which indicates they are not completely prepared to integrate constructionism in the classroom.

When assessing the items on implementation, educators identified insufficient time, financial resources to acquire technology, and especially lack of preparation as the primary challenges that may inhibit integrating coding instruction in the classroom. The latter reasoning garnered the most attention. Implementation possibilities and challenges are integral to evaluate because a sole assessment of pedagogical and epistemological philosophies are not necessarily sufficient to determine whether incorporating constructionism and coding instruction in the classroom is feasible. An affirmative mindset *and* proper preparation and sufficient resources are proportionately critical to effectuate this pedagogical design.

CHAPTER 5

DISCUSSION AND IMPLICATIONS

Study Background

The purpose of this research was to understand what teachers, administrators, and STEM or technology coordinators understand about coding, assess their perceptions on coding instruction's impact on academic achievement, occupational preparedness, and its ability to diminish the disparity in STEM academic and occupational representation, and evaluate if there is a link between educator epistemology and pedagogy with the potential to integrate coding instruction and constructionism in the classroom.

The study was conducted through a quantitative lens and facilitated by a Likert-based survey instrumentation. The survey's chief blocks corresponded to the three distinct research questions. K–12 teachers, administrators, and STEM or technology coordinators from the ADLA participated, and the resulting data were evaluated with descriptive and inferential statistics.

Discussion of Findings

Finding 1

The data demonstrated the survey population's belief that coding is valuable, which corroborates Falloon's (2016) stance that coding inculcates computational and higher order thinking skills. Furthermore, Rich et al.'s (2017) research is validated as well because they indicated one of the most powerful motivating drivers to insert coding instruction in the curriculum is its capacity to develop problem-solving skills. The data also demonstrated participants possessed a general understanding of what coding entails; however, they were not as familiar with the nuances and details of coding because they had not been exposed to its instruction or applications previously. Analysis of all three sets of subgroups (teachers, administrators, and participants with multiple roles; female and male participants; elementary, middle school, high school, and multiple cluster teachers) confirmed similar results. Each subgroup's perception of their nuanced understanding of coding decreased relative to what coding generally entails. This is not unexpected, however, and to best facilitate its instruction in the classroom, teachers and administrators require training. Moreover, data from the entire pool of participants emphasized students' willingness and eagerness to learn coding and affirmed instruction should not be restricted to students who intend to pursue such fields as occupations. This suggests all students should be exposed to coding instruction, which corroborates Falloon's (2016) assertion that coding in the curriculum prepares *all* students for future citizenship, and it should not be restricted to only those who intend to pursue technology related positions.

Additionally, according to the data on the entire survey sample, female interest in STEM subjects compared to boys was higher relative to different ethnic groups; however, according to King (2015), this interest dissipates as they become older. Coding instruction, nevertheless, has the potential to remedy this affect (King, 2015). Subgroup responses based on position (teachers, administrators, participants with multiple roles) lie in the middle of the scale, indicating a lack of firm perception one way or another. Male and female respondents, as well as the subgroups based on grade cluster (elementary, middle, high, and teachers in different bands), replied to this query in a surprisingly similar way. The reasoning undergirding these responses would need to be further investigated to provide an acceptable explanation. The survey item on female involvement in coding was also met with some uncertainty by the total sample of participants, which paralleled all the teachers', administrators', and participants with multiple roles' general responses. The administrator subgroup, however, *somewhat agreed* with this statement indicating their belief that females are less involved. Those who strongly or *somewhat agreed* opined societal factors and less interest are primary reasons for diminished involvement among girls. Nonetheless, the general consensus was uncertainty demonstrated by the predominant mode of *neither agree nor disagree*. This may indicate girls are not perceived to be as involved or interested as boys, but no firm determination could be made. The potential explanations for those who did believe in discrepant interest and involvement between males and females may be societal perceptions, which suggest girls and coding are not compatible or that females have diminished access to coding.

The entire sample population's data also indicated female representation in STEM oriented occupations was similarly deficient to that of ethnic minorities and that this discrepancy was greater relative to the difference in interest between females and males. Teachers, administrators, and multiple role participants did not strongly believe in the presence of a disparity; however, there was some agreement. Male and especially female teachers, however, quite profoundly asserted an imparity exists. The perceptions of those who are directly impacted by a phenomenon in society are especially powerful and revealing. These sentiments were also voiced by the elementary, middle school, high school, and multiple cluster teachers. This simply yet powerfully suggests schools should make a concerted effort to encourage female involvement in STEM careers. Coding can serve as one channel to accomplish this.

These pieces of data are in line with the predominant literature. In fact, Master and Meltzoff (2017) stated women are less likely to hold STEM degrees or careers. Furthermore, the NCES (2018) stated 19% of students graduating with a bachelor's degree in CS are female.

Moreover, less than one third of employees in technology firms are female (Smith, 2016). Master and Meltzoff (2017) posited this reality is attributable to general stereotypes regarding capability. There is a perception that boys are predisposed to succeeding in science and math-based courses relative to females (Master & Meltzoff, 2017). However, data from the survey's sample population disagreed with this sentiment because they asserted female and male students can equally succeed in coding courses and that there is no gender-based distinction in competence. This outlook was shared by every single subdivision in all three sets of groupings. Vegas and Fowler (2020) suggested the identical assertion because success in CS courses or occupations is not rooted in some sort of preordained gendered capability. Rather, ability is connected to access (Vegas & Fowler, 2020), and enhanced opportunity or exposure results in success. There are multiple lines of data from the survey paralleling the literature, suggesting there is no distinction in capability between genders.

Similar lines of inquiries were posed regarding different ethnic groups. For instance, the teacher, administrator, and multiple role participants did not *strongly* believe there was disproportionality in interest in K–12 STEM subjects between different ethnic groups nor was there a strong sense of an imparity in representation in different occupations. To be sure, there was some agreement with the statement; however, it did not necessarily harmonize with the prevailing literature or the data from the total sample population. The male and female respondents believed there is a difference in interest between different ethnic groups, although males were less affirmative with their responses. Moreover, both groups unequivocally asserted there is lack of representation among different ethnic groups in the STEM occupational fields. The four grade cluster groups, similarly to the first set of subgroups, did not strongly believe there was a divide in STEM interest between different ethnic groups; however, their response to the lack of representation in STEM occupations coalesced with females and males. This can indicate ethnic groups do not lack interest; however, they are not equitably represented in STEM occupations. Early access to coding in the classroom can diminish this imparity and provide a more robust pipeline to STEM occupations.

Moreover, the total sample population opined there is disproportionate access to technology in Catholic schools in the ADLA and its adverse impact on different ethnic groups. This population group was additionally of the mindset that certain ethnic groups lack interest and representation in STEM academic subjects and occupations. The entire sample population's and various subgroups' beliefs regarding the lack of diversity in STEM occupational fields coincides with the prevailing research underscoring the identical sentiment for ethnic minorities (Blustein et al., 2013). In fact, Native American, Black, and Latino students are less exposed to CS instruction relative to their peers (K12 Computer Science, n.d.). Calabrese-Barton and Tan (2018) underscored this sentiment as well, asserting students of color lack access to STEM resources and opportunities for exposure. These students face numerous obstacles preventing access to invaluable STEM resources (Calabrese-Barton & Tan, 2018), which manifests itself in decreased graduation rates in such fields. For instance, the NCES (2018) stated 12.7% of undergraduates in STEM fields were Hispanic and 8.5% were Black in 2016–2017.

These disparities are regrettably rooted in the U.S.'s structural dynamics—slavery, racism, and segregation (Avendano et al., 2018). Such inequities ultimately manifested in the educational system as well, which Sparks and Pole (2019) suggested resulted in a reduced number of ethnic minorities entering STEM fields. This substantiates Smith's (2016) assertion that only 3% of employees in technology firms are Black. However, affording opportunities for digital participation to marginalized groups enhances their sense of empowerment (Hagge, 2018), which can improve the diversity in STEM academic and occupational fields among ethnic minorities. Naturally, providing STEM and coding opportunities to traditionally marginalized populations will inevitably enhance interest in academic opportunities

In addition to an evaluation of the entire sample data and on the aforementioned three sets of subgroups, I conducted descriptive and inferential analysis on five subgroups based on constructs, which included elementary, middle, and high schools, participants who teach in multiple clusters, and administrators. The generated data yielded invaluable information.

The first research question's construct consisted of five items gauging participants' understanding of coding The Cronbach's alpha was .84, which exceeded the .70 threshold for reliability, thereby indicating sufficient interrelatedness between the items. The item on participants' general understanding of coding generated the lowest mean and thus indicates a certain comfort level with what it represents; however, their more nuanced understanding of coding paled in comparison as it had a significantly higher mean. This was the case across all other analyses, and this is understandable because coding is studied by a narrow segment of the population. Furthermore, all groups' perceived ability to instruct block- and syntax-based coding also generated high means indicating their discomfort, which once again is identical to the data from the other sets of assessments. These deficiencies highlight the need to allocate resources for professional development sessions. Teachers should be educated on what coding entails, be trained on how to use coding applications and other resources and develop the capacity to integrate its instruction in the curriculum. These sentiments were emphasized by Culp et al.

(2003) as they discussed a study conducted by the NCES, which demonstrated only 20% of teachers could use educational technology. Furthermore, of potentially even greater import, a report from the National Association of State Boards of Education linked professional development and equity in the classroom, suggesting only through such training can student progress and learning truly take place (Culp et al., 2003). Such reports emphasize the need to engender a population of teachers who can use technology in the classroom (Culp et al., 2003).

In totality, data for the first construct signified participants' general lack of understanding of coding; however, the multiple cluster group, comprised of some individuals who taught CS or technology classes, demonstrated a relatively greater level of comfort as indicated by their lower mean figures. The inferential data were in harmony with these descriptive assessments. The Welch's *F* ratio and *p* value showed statistical significance in the difference between means, and the Games-Howell post-hoc test verified this specifically between the multiple cluster group and elementary school teachers in addition to the multiple cluster group and administrators. Moreover, these data were in consonance with the analysis based on participants' roles because the subgroup of multiple roles was comprised of the same individuals in the multiple cluster group who instructed technology and CS. The data on males also signified a heightened sense of efficacy when instructing coding. Although the sample size of males (43) was much less than females (141), the data nevertheless may be attributed to the sentiments from the prevailing literature, which underscored the factors that have created a discrepancy between male and female interest and involvement in coding and STEM subjects. The perceived lack of nuanced understanding by various subgroups suggests the need to invest in resources to educate teachers and administrators about coding and train them to facilitate its instruction.

The first research question attempted to gauge what participants understood about coding and although the survey block asked ancillary items regarding involvement and interest in coding among female students and different ethnic groups, the primary questions' data demonstrated a a lack of understanding especially as it pertains to its nuances and a clear inability to impart its instruction in the classroom. To integrate coding in the curriculum, teachers must have the content and pedagogical background to do so. It begins with an understanding of coding, followed by learning how to code. It is equally crucial to learn how to teach coding and simultaneously inspire student interest and passion toward it. There are numerous coding applications and websites teaching students both rudimentary and more advanced coding, but educators must first form the foundation to facilitate its instruction adeptly.

Finding 2

Data from the total sample population indicated the perception that coding is a unique literacy, its instruction augments student achievement in different courses, it cultivates skills such as problem solving, logical reasoning, creativity, and collaboration, it enhances interest in STEM subjects among females and ethnic minorities, and it better positions these population groups to acquire STEM related positions in the future, which will diversify the representation in these fields. These affirmations from the total sample were also corroborated by the three sets of subgroups. All subgroups from the three sets of analyzed data believed coding can develop logical thinking, problem solving, creativity, and to a lesser degree, collaboration. Quite comparably, all three sets of subgroups' data suggested coding enhances student achievement in different courses, it enhances the potential to secure technology-specific positions, and it mitigates the inequality in the representation of STEM occupations between females and males and different ethnic groups. However, there were some areas where support was less. For instance, the male and high school subgroups, did not as ardently believe in coding instruction's ability to diminish the gap between males' and females' and different ethnic groups' representation in STEM occupations. Furthermore, males, females, elementary, middle school, high school, and participants in multiple clusters did not as strongly believe in coding's capacity to prepare students to obtain positions beyond the technology sector. However, all the subgroups fervently believed in coding's ability to prepare students to undertake technology-related positions.

These data points run parallel to the prevailing literature. For instance, Alexiou-Ray et al. (2018) asserted coding is a unique literacy in the CS field. Resnick and Siegal (2015) appended this statement by positing coding is a novel literacy allowing students to express ideas. Bers (2018) even suggested coding is not restricted to solely CS; rather, coding extended beyond this field and evolved into its own literacy. Such assertions validate the perceptions of participants who suggested coding is a distinctive literacy. Furthermore, Mason and Rich (2020) lent credence to the data indicating coding instruction enhances student achievement in different classes by underscoring studies demonstrating improvement in math and problem-solving abilities. Calder's (2018) research underscored identical contentions by showing coding's ability advancing the grasp of mathematical concepts, including angles and measurements, while Duncan et al. (2014) identified the heightened comprehension of coordinate systems and negative numbers. Apart from mathematics, Popat and Starkey (2019) cited the improved grasp of concepts in art history when coding was integrated in the classroom. Furthermore, Cakir and Guven (2019) emphasized the achievement in science courses that infused coding by describing students’ grasp of the concept of pulse in the circulatory system through a coded robot. Hutchison et al. (2015) also contributed to this general position by suggesting coding develops early childhood literacy, leading to coherent writing and expression of ideas, which will facilitate effective reading, speaking, and writing at the higher education levels.

In addition to the perception of enhanced achievement in various courses, the data on the entire sample population also affirmed the fostering of various general skills such as problem-solving, collaboration, and creativity. Calder’s (2010), Falloon’s (2016), and Popat’s and Starkey’s (2019) research support the respondents’ observations with their suggestion that coding facilitates collaborative work and logical thinking. Resnick and Siegel (2015) also suggested students learn to work collaboratively, develop creativity, and reason in a logical manner. Project Tomorrow (2018) additionally suggested the advancement of creative thought as a byproduct of coding. An example of creativity and coding regarded the HIVE Network in Chicago, which encourages STEM development in youth. One of the applications the youth created is called RideW/Me, which helped students find means of travel to extracurricular activities due to expensive transportation costs (Baker-Doyle, 2018). These opportunities developed civic activism and leadership within students (Baker-Doyle, 2018). Tuomi et al. (2018) provided a summation of these sentiments: “Coding skills are seen as a combination of problem-solving, logical thinking, computational thinking, and design skills” (p. 420).

Data from the survey’s entire population additionally firmly indicated coding better primes students to acquire technology and non-technology related positions, such as in medicine, retail, and government, in the future. Rich et al.’s (2017) study accentuated the same viewpoints because educators in their study offered occupational preparedness as a chief rationale for exposing students to coding instruction. Soykan and Kanbul (2018) contended exposure to coding from an early age will prepare students for any 21st century career, while Tuomi et al. (2018) asserted coding is crucial for various forthcoming occupations. Likewise, Farooq et al. (2012) stated coding is applicable in diverse domains of life. Moreover, Estapa et al. (2018) suggested coding instruction assists students in thinking like mathematicians, engineers, and scientists, and according to CODE.org (n.d.-b), such abilities will provide the requisite foundation to find success in future occupations.

Hutchison et al. (2015) focused on technology related positions and stated coding is a literacy used by fields such as computer engineering, game design, and graphic design. This is important because the U.S. Bureau of Labor Statistics projected an increase in demand for information technology positions by 12.5% from 2014 to 2024 (Bers, 2018). However, the supply of individuals able to fulfill these positions remains insufficient (King, 2015; Rich et al., 2020). Therefore, as suggested by prominent minds who have studied coding's impact on students, it is vital to expose them to its instruction. This will consequently bear benefits to the individual, various industries that integrate coding, and society.

Collaboration, creativity, problem-solving, and enhanced student engagement and achievement in different courses were noted by the total sample population as some of the byproducts of coding instruction, which parallel constructionism's chief contentions. Papert promoted a classroom environment whereby students learn and apply various skills and concepts in mathematics, science, and management (Stager, 2010) with the use of technology. For instance, Harel and Papert (1990) highlighted an example demonstrating the significantly enhanced achievement, motivation, and interest in fractions because of integrating the Logo programming language in the classroom. Prior to this modified environment, students struggled mightily and showed apathy toward the subject matter (Harel & Papert, 1990). Noss and Clayson (2015) provided additional fodder to this contention, positing integration of science and mathematics concepts during the coding process enhances students' problem-solving capacities. Furthermore, collaboration is enhanced as students take individual and collective responsibility to achieve a lesson's objectives (Noss & Clayson, 2015). In addition to collaboration, Alanazi (2016) declared creation of tangible artefacts lies at the core of constructionism. Rob and Rob (2018) appended this statement, asserting creating products and communicating the entailed work in a collaborative setting are critical components of student learning.

Data from the whole sample furthermore declared coding instruction will unlock opportunities for STEM careers for females and ethnic minorities. However, stereotypes and gender biases proclaiming suitability for males have constricted opportunities for females, yet infusing coding instruction in the curriculum will enhance female self-efficacy and thereby diminish the gap in representation (Mason & Rich, 2020). King (2015) suggested coding applications as an avenue for enhanced female engagement. Gunbatar and Karalar (2018) furthered these points by cogently asserting female self-efficacy and positive perception toward coding can subsequently lead to career choices in the field of CS. Duncan et al. (2014) buttressed these points by stating exposure to coding in the middle school can bolster female confidence in high school, which will in turn increase the prospect of a path in CS in higher education. These steps are paramount because career aspirations begin to formulate before high school (Duncan et al., 2014).

Participants also believed integration of coding instruction improved the prospects of ethnic minority groups entering the STEM occupational fields. Lane et al. (2019) highlighted this position, as they attributed the current disparity in representation to factors such as inadequate student recruitment, homogenous faculty, and discrimination in education and in the workplace. Paired programming is one means to ameliorate this predicament (Lane et al., 2019), affording students the opportunity to develop invaluable skills such as problem solving, collaboration, and critical thought. Tissenbaum and Ottenbreit-Leftwich (2020) advanced this general point by stating knowledge of coding is paramount to the success of female and minority students in the CS field and this opportunity must be provided in K–12 schooling. Tran (2019) highlighted an example demonstrating the potential behind these contentions. An academic program from two school districts offered coding instruction to underrepresented populations, including Latino and socioeconomically disadvantaged students (Tran, 2019). Consequently, this developed skills sought by employers, including communication and collaborative abilities (Tran, 2019).

Such opportunities, especially for historically marginalized populations, are especially crucial. It will produce a new generation of students able to meet the demands of current and future workplaces, and thus diversify STEM representation. However, access is critical; students who are not afforded the opportunity to code will have one less critical avenue to develop various skills that will benefit them in the near and long terms. The perceptions of the total sample population and the three sets of subgroups along with the literature underscore the numerous benefits resulting from coding instruction in the classroom. However, a shift in traditional pedagogy and even a restructuring of the curriculum are requisite to effectuate this objective.

The descriptive and inferential data on construct two, which evaluated perceptions regarding coding instruction's influence on student achievement had a sufficient Cronbach's alpha, which exemplified interrelatedness between the five items. The means for all the items, exhibited firm support for coding's capacity to augment student achievement in different subjects, inculcate skills such as logical thinking and problem solving, and enhance student engagement.

As the data were further analyzed, perceptions of each of the five subgroups demonstrated strong support for all the items. The multiple cluster group served as the most fervent of the bunch, while the high school group showed the least enthusiasm toward coding's capability to effect these changes within students. Indeed, there was some variance in the data, which was assessed with the Welch test. Its F ratio of 2.97 and p value of.024 demonstrated there was at least one statistically significant difference in means between the subgroups. Although the descriptive data demonstrated the overall perception that coding positively impacts student achievement, there naturally are differences. The further the F ratio is from 1.00, the larger the difference in means between groups (P. Ruengvirayudh, personal communication, March 14, 2022). Although 2.97 is greater than 1.00, the difference is relatively close, which is in consonance with the one statistically significant difference as noted by the Games-Howell post-hoc test. The p value between the multiple cluster and high school group was .010, which was not unanticipated as these two groups represented the two extremes of the range in means. Other than the statistically significant difference in means between the two groups, which, to be sure, did not indicate the high school group was not supportive, data on the perceptions of all other groups related to coding's impact on academic achievement did not reveal a statistically significant difference in means. This suggests the other subgroups were relatively close in perception regarding coding's influence on academic achievement. Although there was a statistical difference between the high school and multiple cluster groups, the overarching evidence demonstrates coding is beneficial for student achievement. Thus, this serves as one potent reason to integrate coding in the classroom.

The data lucidly demonstrated participants in every set of analysis believe coding instruction can enhance student achievement and engagement and develop logical thinking, critical thinking, collaboration, and creativity. These data are crucial as they not only provide a firm rationale for teachers to commit to learning how to code and instruct it, but it moreover offers a potent motivation for administrators to invest in the requisite resources to integrate it into the curriculum.

The third construct, which was comprised of three items, evaluated participants' perception on coding's impact upon occupational preparedness. The Cronbach's alpha of.75 signified interrelatedness between the items. Furthermore, participants believed in coding's capacity to positively affect the prospect of obtaining technology related positions. When assessing data from the five subgroups, once again, the multiple cluster band generated the smallest mean, which exhibits a firm belief in coding's eventual impact on students' occupational preparedness. This pattern of effusive support for coding is partially attributed to the composition of this subgroup. Furthermore, as was noted in the previous construct's data, the high school group's responses produced the highest mean. The other three groups' means were relatively close to each other. Although data from the high school subgroup mean illustrated a relative belief in coding's impact on preparing students for employment, it would be worthwhile to investigate the reasoning for their relatively depressed measures.

Inferential data on these subgroups corroborated the descriptive indicators. The Welch *F* ratio of 2.75 and *p* value of .034 suggested at least one statistically significant difference in mean and the Games-Howell post-hoc test indeed identified it to be between the high school and multiple cluster group. This provides additional fodder to the need to conduct further investigation to explain why the high school teachers do not consider coding to be as influential, especially because students in that age group are closer to employment. This by no means suggests only the multiple cluster group believed coding assists in occupational preparedness; all subgroups' data signified varying levels of agreement, but the multiple cluster group was the most supportive. If members of the other subgroups were exposed to coding instruction and understood its benefits, it would be reasonable to assume their support for its instruction would also increase. Thus, it is imperative to exert resources to these ends to better prepare students for future occupations.

Data from all points of analyses on Research Question 2, including Construct 3, demonstrate the perception that coding is valuable to occupational preparedness, albeit its perceived diminished influence on preparing students to obtain positions beyond the technology sector. The literature contends coding develops assorted skills that are imperative across sectors; however, even if it solely better prepares students for employment in the technology or STEM fields, it is yet another firm justification to integrate its instruction in the Catholic school curriculum.

The fourth construct, which was composed of four items, evaluated whether participants perceived coding instruction to enhance female and different ethnic groups' interest in K–12 STEM subjects and representation in STEM occupations. The Cronbach's alpha for the four questions assessing these issues was .94, showing a substantial degree of interconnectivity. The means for the items demonstrated the perception that coding can contract the inequality between female and male interest and representation slightly more so than the imparity among different ethnic groups. Albeit the slight difference, it is curious to understand why coding was believed to impact female interest and representation more than different ethnic groups. It may be due to the perception that female interest and representation is a relatively greater problem and thus coding can help mitigate it. Additional research is necessary to explicate this modest difference. Furthermore, quite similarly to the data from the first three constructs, the multiple cluster group's mean exhibited the greatest confidence in coding's capacity to effect change, which in this case referred to enhancing interest in STEM and diversify its occupational representation, while the high school group viewed it with a bit more reservation. The elementary, middle, and administrator subdivisions were of the belief that coding instruction does assist with these issues, but their perceptions lay between the two most extreme groups. Nevertheless, ultimately, although there was discrepancy in perceptions, all subgroups believed to varying degrees that coding augments interest and representation in STEM among females and different ethnic groups.

Although the descriptive data showed a modest contrast in perceptions, the inferential data determined whether there was no statistically significant difference in means between the groups. The Welch F ratio of 1.58 and p value of .188 signified this point and consequently, a post-hoc test was not conducted. To be sure, this does imply there is no variance between the subgroups; rather, it highlights the fact that there is difference rising to the level of statistical significance.

All subgroup analyses on coding's impact on STEM representation demonstrated the belief that it can diminish the imparity in representation, both between females and males and among different ethnic groups. The male and high school subgroups, however, showed less zeal toward this sentiment, while the multiple cluster group demonstrated the greatest degree of fervor. Nevertheless, even when considering these relatively minor discrepancies, it still demonstrates the prevailing perception that coding can positively influence interest and representation among groups that have been traditionally marginalized from such fields. Consequently, this serves as one additional rationale to integrate coding instruction in the classroom.

Finding 3

The survey population's analysis of pedagogical and epistemological philosophies was imperative in determining the feasibility of coding's and constructionism's applications in the classroom. Indeed, Chan (2004) and Schommer-Aikins (2004) explained studies on epistemological beliefs cited a connection between it and teaching and learning. Data from the entire sample, along with all the three sets' subgroups, considered direct instruction to possess some merit, which Popat and Starkey (2019) deemed as important because educators can facilitate development of invaluable skills through well-tailored activities. Furthermore, the entire population's data showed respondents' prevailing convictions that pupil-oriented instruction underscoring enhanced student control, collaborative opportunities, and formulation of knowledge through learned experiences serve as more effective means to education. Similar attitudes were shared regarding the importance of student collaboration and communication as well as constructing knowledge through learning experiences among all the subgroups analyzed, including teachers, administrators, female teachers, and participants in the elementary cluster.

Data from the full spectrum of participants additionally *somewhat agreed* with the notion that technology is a focal tool in the classroom, which may indicate some uneasiness with highlighting technology in the classroom. This was the case as well with all the analyzed subgroups, which poses as a challenge to integrate coding and constructionism in the classroom. Technology is an essential tool of coding and a chief tenet of constructionism, thus its consistent use is necessary. To be sure, the response was nevertheless generally in favor of using technology in the classroom; however, the data indicated a modicum of hesitancy, which can potentially serve as an impediment to employing coding and constructionism in the classroom. A more fervent response may be necessary to effectuate the integration of constructionism and coding in the classroom. Constructionism is animated by digital technologies in the classroom (Noss & Clayson, 2015); therefore, educators or administrators who exhibit some hesitancy may consequently be less inclined to adopt constructionism as a pedagogical medium for student education. The motives for this relative timidity, which may include a sense of limited self-efficacy with the use of technology or student misuse of a device, must also be analyzed to assess the possibility of allaying concerns and preparing educators to use technology in the classroom to meet the needs of students. After all, educators possess the capacity to create an organized and collaborative setting to meet the forthcoming needs of students (Stager, 2010).

The overall sample population's data additionally suggested imparting information and thereafter assessing student comprehension is not deemed as a constructive pedagogical strategy. All the sets of subgroups ranked this statement approximately in the center of the Likert scale. This indicates they are not completely in favor of this strategy, yet they did not absolutely dismiss it either. Another question prompted participants to respond to whether teaching is presenting or explaining the subject matter. There was more agreement with this statement compared to the previously mentioned item; however, it was still relatively tepid. These stances can serve as potential roadblocks to the application of constructionism in the classroom. Although teachers play a vital role in a constructionist academic setting by explaining content, this framework minimizes the centrality of the teacher and instead uplifts the role of the student in the classroom. Professional development and training, in addition to enhancing content knowledge of coding, can provide pedagogical tools to help teachers seamlessly integrate coding and constructionism in the classroom. Knowledge is critical here; if teachers have the necessary tools to impart coding's instruction, it will raise the prospects of it becoming a reality.

This constructionist viewpoint is in line with Popat's and Starkey's (2019) assertion that educators should afford students greater control over the learning process. This can be made possible by integrating collaborative opportunities. Likewise, Falloon (2016), Noss and Clayson (2015), and Rob and Rob (2018) emphasized the advancement of collaboration as one result of coding. Participants validated these assertions by stating these two skills are significant components in the learning process. The data resulting from querying perceptions on the possibility of integrating constructionism in the classroom were crucial. Data from both the total sample population and the various subgroups indicated constructionism, albeit its challenges, is possible to integrate in the classroom. This demonstrates a willingness on the part of educators to highlight the core features of this framework to develop various skills within students.

The varying points in data for the third research question and its nexus with the scholarly literature yield promise to create classrooms within the ADLA that are undergirded by the constructionist pedagogical foundation and coding instruction.

The tenets of constructionism are in harmony with the different points of data for the third research question. Harel and Papert (1990) contended the use of technology supports teaching, learning, and is one means of expression. Stager (2001) appended this assertion by suggesting computers become the medium for creative thought and expression in the form of video game programming, construction of simulations, writing and storage of journals, and production of animated films. Rob and Rob (2018) provided additional fodder to these contentions by explaining the evolved structure of a classroom highlighted by active student engagement and educator promotion of higher-level learning. In this constructionist setting, unexpected knowledge will emerge due to greater student control, engagement, and incentivization (Badilla-Saxe, 2010). Instead of merely being receptacles of information (Rob & Rob, 2018) and learning by reciting or recounting information (Stager, 2001), students construct knowledge due to active immersion (Alanazi, 2016). This environment upends traditional models of teacher pedagogy solely focused on direct instruction and assessment of student comprehension (Noss & Clayson, 2015).

Although the data on pedagogy and epistemology generated auspicious results, when prompted to respond to implementation questions, participants indicated a modicum of reservation. For instance, when asked whether coding instruction should be required, respondents did not entirely concur. Although the predominant response choice was *somewhat agree*, data indicated there were some respondents who *somewhat disagreed* with that statement, while others deemed coding instruction to be superfluous. Additionally, the respondents were not necessarily prepared to infuse coding instruction in the curriculum or able to prepare coding suitable activities relative to their students' grade level. There was a slightly more positive response when asked whether coding should be a standalone subject and whether its instruction should be enmeshed with other courses such as mathematics, language, and science. Furthermore, participants were generally unprepared to impart its instruction, but particularly syntax-based coding. Ultimately, this overarching source of hesitation may be based on a lack of information regarding coding instruction and its benefits, but even more so rooted in the educators' unpreparedness to infuse its instruction in the curriculum. Thus, data on implementation are not necessarily in consonance with data on pedagogy and epistemology. This may suggest that, although educators believe in the potential for constructionism and coding, they are not prepared to integrate it in the curriculum.

These sentiments were abundantly echoed by the data emanating from all the subgroups. The primary challenge selected by participants from both the perspective of the teacher and administrator was lack of teacher preparation, which corroborates various pieces of literature. Teachers strongly believed they are ill prepared to immerse coding in the curriculum. To incorporate its instruction, teachers must be adept at not only effectively using coding apps and resources but also are competent in facilitating its instruction. Lack of resources or insufficient time were some additional cited reasons; however, the predominant choice dealt with unpreparedness. This very point is explained in depth in the prevailing literature, and it undergirds the contention that proper training is necessary to equip teachers and administrators with the tools to adequately impart and integrate coding's instruction. Rich et al. (2020) and Harlow et al. (2016) highlighted this contention by stating coding instruction is not imparted in most elementary schools in the United States, and one of the chief impediments was the lack of teacher preparation (Rich et al., 2020). Kong and Wong (2017) furthered this contention stating teachers considered themselves less technologically savvy relative to their pupils, they did not possess the requisite content knowledge, and they lacked the pedagogical foundation to integrate coding. Indeed, 75% of educators either have no coding experience or very limited prior exposure to it.

Additional data demonstrating the perceived sense of limited efficacy or preparation regarded the instruction of block- and syntax-based coding. Generally, most subgroups relayed their inability to teach either form of coding; however, the multiple role participants, male, and multiple cluster teacher subgroups revealed a marginally heightened sense of efficacy. The male perception may be rooted in their increased exposure to coding, which is supported by literature demonstrating the historical inequitable access to coding and STEM resources, while the multiple cluster teacher subgroup is comprised of some CS and technology teachers. Moreover, when prompted to respond to whether they can create coding activities at the appropriate level and integrate coding into the content curriculum, the subgroups' responses were resoundingly negative. The multiple role, male, and manifold cluster participants perceived themselves at being slightly more prepared, yet the overarching sentiment in these three subgroups was discomfort. The reason why they fared only marginally better may be because they are composed of individuals who teach CS or technology and males who may have been more exposed to coding instruction or have some familiarity with it.

Another crucial question urged participants to opine on the need to restructure the curriculum to include coding. All the subgroups' responses expressed moderate agreement with this statement. The multiple cluster subgroup revealed the strongest concurrence with this statement, which may be due to the presence of some CS or technology teachers. This question was posed to provide an additional piece of data to clarify whether the current structure can include coding or if there is need to enact a significant change to accommodate the instruction of this literacy. The overall modest agreement with this statement does bode well for coding's integration. If teachers and administrators are willing to create structural change, it enhances the potential to integrate coding.

Additional points in the data stemming from the entire sample population and the prevailing literature corresponded. For instance, Rich et al. (2017) identified mathematics as the predominant course with which coding was integrated; however, they additionally noted educators wove coding in other subject areas as well. Integrating coding in different courses can resolve the issue of finite instructional or curriculum time because a separate course would not be dedicated to its instruction (Project Tomorrow, 2018).

These powerful contentions are in accord with the data—ADLA teachers are generally ill-equipped to immerse coding instruction due to lack of knowledge and preparation. Furthermore, although the data demonstrated a shift away from teacher-focused instruction and an agreement that constructionism is feasible to employ in the classroom, there may nevertheless be a need to expose educators to the constructionist pedagogy as one means to facilitate coding instruction in the classroom. Barr and Stephenson (2011) and Bell (2016) underscored the importance of a shift in pedagogical philosophy as well. This may very well serve as the first step in moving toward a curriculum with coding even before preparing teachers to facilitate the instruction. Once a foundation to implement coding instruction is established, educators can subsequently undergo preparation to mediate its instruction. Mason and Rich (2019) and Seow et al. (2017) suggested preservice programs and professional development sessions can help forthcoming and existing educators alike. With the requisite foundation and tools in place, educators can facilitate coding instruction in the classroom.

Descriptive data were also conducted on the final set of subgroup analysis, which encompassed the most wide-ranging number of groups comprised of elementary, middle, high, and multiple cluster teachers and administrators. A construct was not created for this set of items because it did not meet the .70 Cronbach's alpha threshold. Nevertheless, it was crucial to gauge participants' perceptions on epistemology and pedagogy to ascertain whether it is in line with integrating constructionism and coding in the classroom.

Because constructionism highlights collaboration and communication in the classroom, participants were asked whether these are a vital means of learning. Their robust responses generated a mean of 1.23 demonstrated their agreement. Furthermore, participants were queried as to whether the focus of teaching is to help students construct knowledge from learning experiences instead of imparting knowledge, which is another chief principle of the framework, and the data indicated their support for this sentiment as well. The straightforward question asking if constructionism is feasible to apply also received support, albeit less relative to the aforementioned two items. Furthermore, direct instruction was not necessarily perceived to be an effective means to student learning and the item probing perceptions on whether teaching is presenting or explaining the subject matter showed a relative a lack of support for this sentiment. These two pedagogical strategies are not in line with constructionism, further enhancing the prospect of its implementation. However, participants did not necessarily believe technology should be an integral tool in the classroom nor did they assert students should have greater control in the classroom. These two principles are integral parts of the paradigm; thus, these data present a potential challenge to integrating constructionism.

Constructionism serves as one channel through which coding can be integrated in the classroom. It champions a student-centered environment that inculcates multifarious skills within students using technology and coding. Ultimately, an overarching view of the data suggests support for *some* tenets of constructionism and its integration, but other crucial pieces of this framework, such as allocating greater control to students or uplifting technology's role in the classroom, lacked comparable support. Consequently, teachers and administrators in the ADLA are *not* completely prepared to integrate coding instruction through the constructionist lens. To effectively integrate constructionism or any other paradigm in the classroom, the teacher simply must believe in all its tenets. Because the data did not indicate teachers and administrators in the ADLA are in line with all of constructionism's principles, I suggest integrating constructionism to facilitate coding is not currently feasible in ADLA Catholic schools. Nevertheless, like the need to develop participants' understanding and instruction of coding, efforts can be undertaken to educate teachers on the principles of constructionism and identify means to incorporate the paradigm. Professional development and training are critical to effectuate the integration of constructionism and coding in the classroom and I remain hopeful that this can be achieved in ADLA Catholic schools in the future.

Limitations and Delimitations

Several limitations impacted my research. First, the limited time to gather data presented itself as a challenge. My data gathering process commenced in the summer of 2021 and I began to analyze the data in November 2021. Additional time, however, would have, in all likelihood, yielded a greater amount of data. The second limitation was the response rate of the potential participants. Time is a precious commodity for educators; the daily schedule of teachers and administrators is hectic, which adversely impacted the quantity of data gathered. The third limitation was the lack of curriculum integrating coding instruction in the ADLA. Although some schools infuse a modicum of coding and emphasize STEM content, many archdiocesan schools do not; therefore, the basis of my data was educator and administrator perceptions regarding coding. A future study may conduct a longitudinal study of a school's curriculum with coding to assess its influence on student academic achievement, preparation for various occupations, and its ability to dent the gap in representation in the STEM fields. The fourth limitation regarded the development of the four constructs. I did not validate them empirically using factor analysis; however, their inception was based on what theoretical frameworks and the literature have stated regarding these questions, which provides legitimacy to the constructs. The fifth and quite possibly the most impactful limitation was the global COVID-19 pandemic. It further exacerbated the inherent challenges of the data gathering process. I initially distributed the survey in the beginning of October to provide participants time to acclimate to the new academic year beforehand; although 2 months had passed since the beginning of the academic year, the additional responsibilities and stressors educators shouldered inhibited their inclination to participate in the study. This consequently delayed my ability to analyze the data. The challenges schools experienced also caused certain principals to decline my invitation to participate in the study.

The first delimitation of my study was its focus on the ADLA. Although I originally intended to conduct a national study incorporating dioceses of varying sizes, a narrower study emerged as Dr. Reilly and I discussed its ultimate path. Although the global pandemic had not sprung at the point of our decision, a national study would have presented monumental challenges, especially in terms of data gathering. My study's focus on teachers, administrators, and STEM coordinators was yet another limitation. There are other shareholders in an educational setting, such as parents and students; however, I intended to assess perceptions of this segment of a school's population to determine whether coding instruction is feasible to institute in Catholic school curriculum. Another delimitation was my emphasis on coding instruction. Coding is a part of the larger field of CS, and I decided to center my study on it for two reasons: first, there was some momentum toward its instruction in schools and second, a professional development session in which I participated years ago explained what coding was and highlighted its benefits. As an educator, this planted a seed of interest within me at that moment in time and it would germinate into a dissertation topic in a doctoral program some time later. An additional delimitation involved an intentional focus on specific items from the survey corresponding to each research question when I descriptively analyzed the three sets of subgroups. I believed those questions were the most pertinent to each research question and to make this study manageable given the time parameters set upon the dissertation, I restricted the analysis to certain questions, but they were nevertheless expansive in scope and breadth. Finally, based on consultations with my committee members, I decided to analyze three sets of data based on demographic information. Although there were other points of demographic data I could have evaluated such as race, credentialed and non-credentialed teachers, and length of service in the educational setting, I decided to choose the following three because they were of interest to me. First, I closely analyzed comparisons and differences between teachers, administrators, and participants with multiple roles at their school sites. I furthermore analyzed differences in perceptions between female and male respondents. Converging and diverging perceptions among teachers in different grade clusters, partitioned between elementary school, middle school, high school, and multiple band teachers, were also evaluated. A more expansive analysis, conducted through descriptive and inferential means, focused on elementary, middle, and high school teachers, participants who instruct in multiple grade clusters, and administrators.

Future Research

The original scope of this study intended to gauge the understandings and perceptions of educators and administrators across various dioceses in the nation using a survey-centered methodology. However, given the limitations of this study, a national study was beyond reach. It is nevertheless important for future research to address similar lines of inquiry within various larger and smaller-scaled dioceses in different states to assess whether coding instruction and constructionism are feasible to integrate in Catholic school curriculums.

Another potential study would be narrower by focusing on one diocese as I did, but its methodology would be qualitative. The data would be gathered primarily via interviews with educators and administrators from different schools in the diocese, which would provide a deeper understanding of people's perceptions and provide a firm footing to assess whether coding instruction and constructionism are possible to integrate.

Additional routes for future research include a focus on non-religious private, public, charter, and magnet schools. Qualitative, quantitative, or even a mixed method methodology can be employed to conduct these studies. This would be a colossal undertaking; however, my ultimate objective is to see coding instruction integrated in all schools across the country. All students should have the opportunity to learn this novel literacy and have an additional means to develop invaluable skills. Therefore, evaluating this prospect in all school systems is vital.

There is the potential to continue assessing the current data set from my study. For instance, analyzing alternative demographic data based on race, age, length of service in education, and credential status, may yield invaluable data revealing differing perceptions relative to the subgroups already evaluated in this study.

Implications

This study diminishes the extant gap in data regarding Catholic schools and coding instruction. Even more significantly, with the support of the data based on the perceptions of participants, it espouses the integration of coding instruction in Catholic schools and even a structural shift in the curriculum to ensure its reality. Coding can be integrated through the framework of constructionism, but it requires significant resources to prepare teachers to possibly alter their pedagogical approaches and develop their knowledge of coding and ability facilitate its instruction. The literature suggests numerous benefits including enhanced student academic achievement, the development of invaluable skills such as problem-solving and critical thought within students, and the better positioning of a much wider segment of the student population to not only acquire a broad range of positions, but specifically STEM related ones. Naturally, theoretical, practical, and policy implications present themselves as certain significant changes are requisite to effectuate the immersion of coding instruction.

Theoretical Implications

Constructionist pedagogy is undergirded by the utility of technology to uplift the role of students during the learning process. In this setting, students collaborate and are actively engaged as they learn the curriculum content by creating artifacts. Constructionism's tenets are in accordance with coding instruction, and it thus serves as one viable means to implement its instruction. The data on the entire sample of participants along with the four sets of subgroups exhibited a belief in the possibility of integrating constructionism. This is ample proof that constructionism and coding can be a reality in Catholic school classrooms, and it can serve as an example for other school systems to follow. To accomplish this goal, educators, during a multi-pronged professional development program, can be exposed to constructionism's principles, coding instruction, and their nexus, thereby providing them with the foundation to facilitate their integration in the classroom.

Implications for Practice

Coding's integration will undeniably present challenges and changes in the classroom. Educators will either instruct coding as its own standalone course or integrate it during the facilitation of other courses. Especially with the latter alternative, pedagogical philosophies will be altered as traditional and current teaching strategies will be reformulated to include the integration of this new literacy. In fact, technology will serve as the center of student life in the classroom as they develop vital skills such as critical thought, problem solving, communication, and collaboration.

With such a revolution in classroom structure and pedagogy, teacher commitment, enthusiasm, and preparation are critical. Thus, an organized professional development program is necessary to effectuate coding instruction in each school. The Department of Catholic Schools (DCS), which is part of the ADLA, is decentralized in nature, allowing each school to remain independent and implement its desired programs. This structure has its benefits; Catholic schools in the ADLA possess the latitude to implement programs as they deem fit, thus they can use this flexibility to integrate coding instruction. Schools in other realms, such as in the public sphere, do not necessarily enjoy this type of independence. However, a centralized effort can effectuate this goal more efficiently. This can come in the form of professional development sessions that gather multiple schools at once. Prior to the pandemic, the DCS did just this focusing on one theme or objective each year during professional development sessions. For instance, during one academic year, they focused on preparing educators to integrate mathematics in all course content. A similar effort can take place, but instead emphasize coding instruction. The DCS has many responsibilities, and they take invaluable steps at helping schools in the diocese best serve their families; however, I am placing the onus on them. Their central role and capacity must be used to push the integration of coding forward in Catholic schools. There would need be "buy-in" at this level, and I believe I have the knowledge and positionality to make a sound case. After this point, they can invest in resources to assist schools in understanding what coding entails, explain their various benefits, teach educators and administrators how to use coding applications and resources, and prepare them to integrate it in the curriculum. It is also critical to keep in mind that students often remain in the same Catholic school until the end of middle school; therefore, if Catholic schools integrate coding instruction in the curriculum from K–8, students will receive consistent exposure and thus be better situated to develop invaluable skills and acquire STEM related positions relative to their peers in other schools.

As I experienced various doctoral courses with my fellow cohorts at LMU, we were afforded numerous opportunities to reflect upon our problems of practice and identify the means to rectify it. As part of the LMU Organizational Theory and Change course (McCullough, 2020), the final assignment was to create an organizational plan to resolve our problem of practice and thus establish substantive change in our school setting. Of course, I identified the lack of coding instruction in our Catholic school curriculum as the obstacle and I devised a two-year plan, based upon a theory of organizational change, that would remedy it. I believe this plan can be incorporated in schools moving forward.

Stroh (2015) championed systems change and presented a guide to accomplish it, including analyzing the entire system rather than merely assessing one's role, assisting others in changing, and reconfiguring deeper system structures. In addition to these general guidelines, Stroh presented four concrete stages to effect systemic change: 1) establish a foundation for change and ensure others in the system are prepared to assist in its fruition 2) identify the current reality in the workplace and everyone's role in establishing and perpetuating it 3) select an aspiration or goal that all advocate 4) create the change by focusing on high-leverage interventions, seek assistance and feedback from other constituents, and evolve based on experiences. I can apply this blueprint to integrate coding instruction not only in my school setting at Holy Family Grade School but help drive this forward in Catholic other schools as well.

The change I seek to implement, which foundationally alters pedagogy, must be tailored meticulously and be based on collaborative efforts. Owens and Valesky (2015) contended change in an organization must be planned, involve the entire organization, and enhance the ability of the organization to change to suit current and future needs. To follow the processes for transformation Owens and Valesky (2015) suggested a foundation must be laid initially, which is Stroh's (2015) first stage for change. In line with this first pillar, the benefits of coding should be espoused, and pedagogical alterations should be highlighted. Professional development sessions and first-hand experience with rudimentary coding applications are critical in this first step, which will take place during the first year of my proposed plan. It is beneficial to be presented with data and information regarding coding; however, there will be even greater "buy-in" if teachers bear witness to its advantages. Furthermore, creating a shared understanding that voicing concerns and criticisms are welcome during this process will be critical in producing the change. Indeed, Stroh (2015) asserted, "Creat[ing] a strong and safe container for people to share their different perspectives" (p. 34) is a crucial step in systemic change. As there is progress toward the goal, it is also incumbent upon each educator and administrator to identify their role in the current formulation of the curriculum, which highlights Stroh's second stage in the change process. With all the responsibilities teachers and administrators bear, a pedagogical shift will be replete with challenges; however, to best facilitate the needs of our students in an evolving society, a uniform embrace of this vision is necessary. This underscores Stroh's third element in the change process.

Finally, in line with the final stage in Stroh's (2015) design, educators must devote the necessary time and effort to learn how to embed coding during the instruction of various disciplines. This step will take place during the second year of my plan through engaging and deep professional development sessions and will be by no means simple to achieve because it will necessitate the restructuring of each educator's foundations for teaching. However, I am confident that a firm collaborative culture that prioritizes students' interests will continuously guide the educators' efforts and thus spur this change. Once educators are trained to incorporate coding with the instruction of various traditional disciplines, educators will be equipped with the tools to generate the change.

The parents' role in this equation is of equivalent significance. Once coding has been integrated in the curriculum and students have had the opportunity to become active agents of and co-contributors in their learning, parents must witness the products of our efforts and provide feedback. On a weekly basis, teachers and the principal can email a newsletter to parents and through this medium, parents can be informed about the coding lessons taking place in the classroom. Consequently, parents will be abreast of not only the activities in their child's classroom but will also understand the efforts in all other classrooms. Furthermore, either on a quarterly or semesterly basis, an open-house event solely dedicated to coding can be an additional avenue for communication. It is not only important for educators to communicate with parents, but parental feedback is crucial as well to mold the program to suit the needs of students. Hargreaves and Fullan (2020) asserted practices such as parents' night allowing students to explain their work keeps parents engaged and simultaneously positions them to assist their children in the work they complete.

Finally, and most importantly, assessing this program through the lens of students is of utmost importance. In modern society, technology is an ever-present part of their daily lives, and schools can better reinforce the lessons taught in school through technology. As the STEM director at HFGS, I have witnessed the exuberance of students during our after-school robotics sessions. Students are eager to collaborate and code their robots to perform specific functions. Rather than exposing a very finite number of students to such opportunities, coding can be made available to all by molding the curriculum to integrate coding. Hargreaves and Fullan (2020) lamented the disconnect between pedagogy and curriculum to students' diverse needs and their escalating lack of engagement and alienation in the classroom. According to Hargreaves and Fullan (2020), "Thus, students in 2020 find schooling less and less relevant and interesting compared to previous generations" (p. 332). I contest this reality can be countered with a curriculum that embeds technology through use of coding. The COVID pandemic compelled schools to use technology to communicate with students during distance learning. Although in-person instruction has resumed, this momentum should and must continue. Hargreaves and Fullan (2020) averred schools with a strong foundation for collaborative work provided high- and low-end technological innovations well suited for present and future realities. We can use this momentum, extend our efforts by integrating coding in the curriculum as well, and especially assist schools lacking the resources and knowledge needed to do so. This COVID pandemic inadvertently gave us the platform to build our professional capital, as termed by Hargreaves and Fullan, and as educators be better positioned to serve our students. Indeed, Hargreaves and Fullan suggested digital opportunities can be used to enrich professional capital and community, which will consequently enhance student experience and learning.

Policy Implications

Coding instruction necessitates curriculum alterations and preparation of educators; however, it also presents policy implications at the state level. Although the Common Core Standards dictate curriculum guidelines, education policy remains under the purview of each state. Thus, it is important to educate state officials regarding the benefits of coding instruction and apply pressure upon them to create legislation that provides resources to schools to meet this end. Yadav et al. (2016), for instance, cited the need to change policy and create programs solely focused on CS teaching licensure to equip teachers with the content and pedagogical knowledge to facilitate a CS class effectively. Financial and content-based resources are imperative to equip schools with the money and knowledge needed to achieve this goal. Although my research focused on the ADLA, and private schools do not receive the same type of funding public schools do, my ultimate objective is to see coding's integration in the curricula of all types of schools. To this end, I spoke with California State Senator Anthony J. Portantino (A. Portantino, personal communication, June, 5, 2020) during an LMU course titled "Context and Current Topics in Public Education" (Lavadenz & Kaminski, 2020) while schools remained in distance learning.

I introduced my stance on technology and coding to State Senator Portantino (A. Portantino, personal communication, June 5, 2020) and he responded that these became important given the requirement of distance learning during the pandemic. It especially shone a bright light on the discrepancy in access to resources for students from different socioeconomic backgrounds. This point was noteworthy because disproportionate access was also indicated in the literature and in the data. He additionally referenced the dearth of students who are prepared for coding positions and lamented the lack of female representation in such positions. This paucity was also suggested by the literature and the data. Furthermore, he agreed with my assertion that coding should be implemented in the curriculum to repair the leaky pipeline of qualified individuals who can fulfill the tasks associated with coding positions.

To effect any change, such as in education, policy is needed. State officials such as Senator Portantino are positioned to respond to our voices and implement changes in education because it is under the purview of the state. I am hopeful his stance on technology in the classroom, his understanding of the need to prepare a stronger STEM pipeline, and my communication with him served as a seed that may germinate policy that provides funding for coding programs in schools and curriculum change. Nevertheless, one conversation is insufficient; a concerted and consistent effort is necessary to spread the message and receive legislative support. Although monetary assistance is not the only element involved in this entire equation, it can assist schools in providing professional development sessions to their educators and hiring staff who can impart coding instruction.

A second potential policy implication impacts higher learning institutions. Instead of solely relying on the school site, the diocese, or the district to prepare teachers to incorporate coding instruction in the curriculum, colleges and universities can embed appropriate training in their preparatory programs. One contention of the 2019 State of Computer Science Education Report was the need for institutions to create programs to prepare preservice educators to integrate CS in the curriculum (Code.org Advocacy Coalition, 2019). Mason and Rich (2019) and Seow et al. (2017) corroborated this point by positing preservice teachers and even seasoned ones should have professional development sessions that prepare them to instruct coding.

Cutumisu and Guo (2019) suggested university teacher preparation programs use resources on CODE.org to prepare preservice educators. The impact of such programs was highlighted by Chang's and Peterson's (2018) research explaining how 44 preservice educators, including some who were initially skeptical of coding, understood coding's ability to develop invaluable skills and were prepared to integrate coding.

A final policy implication resulting from my research impacts the ADLA. Although the Catholic schools within the ADLA's auspices function independently and the central body does not prescribe curriculum or any other school related policy, it should nevertheless use its capacity to organize a tiered program that first explains what coding is while highlighting its benefits, next identify and demonstrate various applications and resources for coding, and subsequently teach educators how to initially use it and then how to teach with it. All of this cannot be accomplished in one or even a few professional development sessions; rather, a 2- or even 3-year program is necessary whereby educators from schools within different regions of the ADLA take part in multiple sessions each academic year. My 2-year program explained previously is a prescription for each specific school site; however, a centralized delivery of a program teaching educators how to integrate coding instruction in the curriculum will be more efficient and possibly effective. Each school can then use their learnings and employ it in a fashion that best suits their needs.

Recommendations

The first recommendation is to reassess the reasons why the fifth construct was not created. Additional items need to be formulated, the existing ones should be modified, and critical feedback is necessary to create one construct addressing the potential link between pedagogy and epistemology with the integration of constructionism and coding in the classroom.

The second recommendation centers upon the ADLA's shoulders. They must use their central capacity to provide all our students the opportunity to develop and advance numerous instrumental skills such as problem solving, collaboration, and creativity. The ADLA has a very diverse body of students, and Catholic schools in general pursue social justice aims such as equity and equality. By integrating coding instruction in the curriculum, the ADLA can provide minority students, who have been traditionally marginalized from STEM, with this powerful opportunity and thus come even closer to reaching their social justice aims. Catholic schools pride themselves on reaching every child by facilitating their academic, social, emotional, and spiritual growth. Indeed, by focusing on strategies such as differentiated instruction and offering CS or technology courses in the curriculum, Catholic schools consistently strive to maximize the potential of each student and provide equitable access to opportunities; however, based on the data, there is the perception that there is a divide in STEM and coding opportunities. To remedy this issue and better prepare all Catholic school students for academic and workplace challenges, it is crucial that the ADLA establish a multilayered program to prepare schools to implement coding in their curriculum.

A scaffolded program explaining what coding is, highlights its benefits to student achievement in the near and long terms, trains school personnel on how to use coding applications and other resources, and prepares them to integrate its instruction in the curriculum is crucial so pedagogues do this willingly, effectively, and efficiently. As I have stated previously, "buy-in" is critical among administrators and teachers at each Catholic school. The responsibility subsequently shifts to each school site because they ultimately must integrate the instruction into their daily curriculum. Administrators and educators should coordinate to cohesively incorporate its instruction and routinely reflect upon how the instruction is proceeding to identify areas for growth. The ADLA's representatives should also visit each school site to check their progress and assist with any needs.

Next, public schools and their central districts, charter schools and their charter management organizations, magnet schools, and nonreligious private schools should use a similar blueprint to integrate coding instruction into the curriculum. Of course, the dynamics and needs of each type of school environment will vary; thus, they should customize this program to suit their own needs. Overarching guidelines focused on a gradual tiered program initially emphasizing understanding and then application should be employed.

Fourth, more research is necessary to comprehend a wide variety of issues pertaining to coding. More studies should be conducted in a wide spectrum of schools, including religious, public, charter, and private, to gauge teachers', administrators', students', and parents' perceptions of coding instruction. Further investigation is warranted to explain some of the resulting data from this study. For instance, the ADLA high school teachers' slightly reserved perceptions regarding coding requires additional analysis. Although they generally signaled support toward coding, their responses were not as enthusiastic relative to the other groups, especially the multiple cluster subgroup. The literature emphasizes the importance of integrating coding instruction consistently from K–12 and because high school students are closest to choosing their future sphere of employment, it is significant to evaluate the rationale for this subgroup's perceptions. If high school teachers do not integrate coding instruction or do not do so consistently, it can adversely impact students' interest and continued growth and learning in this area of CS. Ultimately, this may hamper efforts to diversify the STEM pipeline.

Furthermore, schools integrating coding instruction should have their program's effect on student achievement analyzed. Longitudinal studies should be conducted on students consistently exposed to coding instruction to assess its impact on their academic achievement and eventual occupations. When and if coding taking root in K–12 schools, representation in the STEM academic and occupational fields must also be evaluated to determine if it impacted these realities in any manner. Furthermore, the role of higher education schools should also be focused upon as they have a significant impact on preservice teachers' pedagogical underpinnings and content knowledge. Such programs should allocate resources to help teachers understand what coding is and to provide pedagogical tools to assist in facilitating its instruction in the classroom. Universities offering such programs should be studied to assess their impact on the issues raised in this dissertation.

Conclusion

I view coding as an educational leadership channel to advance student academic achievement, facilitate the development of diverse invaluable skills such as problem solving and critical thinking, and advance social justice by diversifying the representation of STEM occupations. If students of all backgrounds and genders are afforded an equitable opportunity to learn coding, it will inevitably lead some to further pursue it in higher education and as a career path. Thus, a more inclusive educational system is fostered as every child is afforded an additional opportunity to succeed in the near and long terms. Not only are such positions well respected and compensated, but a curriculum that integrates coding will create a more diverse pipeline to STEM occupations. Students who do not pursue occupations related to STEM will nevertheless be equipped with applicable skills of utility in many other fields, be it medical, government, retail, or hospitality. As an educator and STEM director, I have the motivation, moral obligation, and the platform to effect this change and assist others in spurring it as well. It is a responsibility I do not take lightly, and I am committed to pursuing it until it is met. This study presented data based on perception of teachers and administrators in K–12 Catholic schools in the ADLA. It provided a wealth of data demonstrating coding's value, especially as it pertains to uplifting our students by equipping them with the tools necessary to succeed in the future. Although integrating coding instruction in the classroom will present challenges, the potential to positively impact students' immediate and future lives are powerful reasons that should serve as an impetus for the ALDA to take action.

APPENDIX A

Formal Request to Conduct Study in the ADLA

Krikor Kiladjian
Loyola Marymount University
1 LMU Drive
Los Angeles, CA 90045

Dear Dr. Galla,

My name is Krikor Kiladjian and I am a doctoral student in LMU's Educational Leadership for Social Justice program. I am currently in the second year of the program and I have begun writing my dissertation, which hopes to center upon coding instruction in Catholic schools in the Archdiocese of Los Angeles. This letter serves as a formal request to conduct the study within schools in the Archdiocese of Los Angeles.

The aim of this research is to understand the perceptions of a comprehensive collection of stakeholders on coding's impact on student achievement in the classroom, occupational preparedness, and its potential to dimmish the gap in access to technology and STEM related content for students from traditionally marginalized populations. Specifically, teachers, administrators, and STEM and technology directors will be asked to participate from Catholic schools in the ADLA via a survey that has approximately 65 questions. The questions will primarily be fixed choice in nature, but some will be open-ended as respondents will be asked to expound upon their attitudes or beliefs.

There is increased prominence accorded to STEM disciplines, including computer science and in particular coding instruction, by educational and legislative leaders. Additionally, there is a growing sentiment among the U.S. public that computer science and coding instruction are essential for future success. Consequently, I would like to amplify this momentum to ensure the fruition of a uniform shift in curriculum that embraces coding in Catholic schools in the ADLA.

My research question is what do teachers, administrators, and parents know about coding and what are their perceptions on its impact on students' academic achievement in the classroom, occupational preparedness, and effect on promoting social justice?

The benefits of the study will highlight a need in our Catholic schools. Coding instruction inculcates various skills such as critical thought, collaboration, and communications and thus, it possesses the potential to enhance student achievement, better prepare our students for forthcoming occupations in a global society and bridge the divide that exists in access to technology and academic and career opportunities for traditionally marginalized groups. The data from this study will be shared with participants and will ideally inform pedagogical decisions of school personnel to meet these ends.

APPENDIX B

Formal Request to Principals to Conduct Study

Krikor Kiladjian
Loyola Marymount University
1 LMU Drive, Los Angeles, CA 90045

Dear Catholic School Principals,

My name is Krikor Kiladjian and I am a doctoral student in Loyola Marymount University's Educational Leadership for Social Justice program. I am currently in the second year of the program, and I have begun writing my dissertation, which centers upon integrating coding instruction in Catholic schools in the Archdiocese of Los Angeles. I have also served as a Catholic school teacher for the past 11 years at Holy Family Grade School in Glendale. As educators, we persistently strive to best prepare our students for the academic and professional challenges they will encounter and it is my respectful, yet firm belief that we can do so by integrating coding instruction in our curriculum. My study is survey based and will seek to understand educators', administrators', and STEM and/or technology coordinators' or directors' comprehension of what coding entails, the academic, vocational, and social justice benefits that derive from its instruction, and the potential means of implementation in the classroom. This letter serves as a humble and kind request to conduct the study within your respective school by sending an approximately 65 fixed-choice question survey, which includes some open-ended questions, to your educators, administrators, and STEM/technology coordinators.

The benefits of the study will highlight a need in our Catholic schools. Coding instruction inculcates various skills such as critical thought, collaboration, and communications; thus, it possesses the potential to enhance student achievement, better prepare our students for forthcoming occupations in a global society and bridge the divide in access to technology and academic and career opportunities for traditionally marginalized groups. The data from this study will ideally inform pedagogical decisions to meet these ends.

APPENDIX C

Primary Investigator's CITI Certificate

Completion Date: 08-Sep-2019
Expiration Date: 06-Sep-2024
Record ID: 33087409

CITI PROGRAM

This is to certify that:

Krikor Kiladjian

Has completed the following CITI Program course:

Social & Behavioral Research - Basic/Refresher (Curriculum Group)
Social & Behavioral Research - Basic/Refresher (Course Learner Group)
1 - Basic Course (Stage)

Under requirements set by:

Loyola Marymount University

CITI
Collaborative Institutional Training Initiative

Verify at www.citiprogram.org/verify/?w7bf1a5c3-740f-4a84-833b-c49fcdeb7673-33087409

APPENDIX D

Chair Professor's NIH Certificate

Certificate of Completion

The National Institutes of Health (NIH) Office of Extramural Research certifies that **Elizabeth Reilly** successfully completed the NIH Web-based training course "Protecting Human Research Participants."

Date of Completion: 09/25/2018

Certification Number: 2959049

NIH National Institutes of Health
Office of Extramural Research

APPENDIX E

IRB Application Cover Sheet

Received____________________

LOYOLA MARYMOUNT UNIVERSITY

Human Subjects Research

APPLICATION TO THE LMU INSTITUTIONAL REVIEW BOARD (IRB)

Principal Investigator (PI): Krikor Kiladjian

Title of Project: Perceptions of Coding Instruction in K–12 Catholic Schools in the Archdiocese of Los Angeles

P.I. Type: (check one) ☐ Faculty X Graduate ☐ Undergraduate ☐ Staff

Department: School of Education, Educational Leadership and Administration

Campus Address: 1 Loyola Dr., Los Angeles, CA 90045

Telephone: 818-307-3339 E-mail: kkiladji@lion.lmu.edu

Faculty Sponsor (if applicable): Elizabeth Reilly

Submission: X New ☐Renewal ☐ Addendum ☐ Staff ☐ Other Previous IRB#:

For evaluation of your project, indicate involvement of any of the following:

- ☐ Audio Recording of subjects
- ☐ Filming or video-recording of subjects
- ☐ Deception
- X Questionnaires or surveys
- ☐ Psychology subject pool
- ☐ Elderly subjects (over 65)
- ☐ Experimental drugs
- ☐ Establishment of a cell line
- ☐ Placebos
- ☐ Non-patient volunteers
- ☐ Minor subjects (younger than 18)
- ☐ Interviews
- ☐ Subjects to be paid
- ☐ Fetal tissue
- ☐ Subjects studied off campus
- ☐ Experimental devices
- ☐ Surgical pathology tissue
- ☐ Patients as subjects

- ☐ Photographing of subjects or artifacts created by subjects
- ☐ Subjects with impaired decision making capacity
- ☐ Economically or Educationally Disadvantaged Persons
- ☐ Approved drugs for "Non-FDA" approved conditions
- ☐ Subjects in Armed Services (Active Duty)
- ☐ Prisoners, parolees, or incarcerated subjects

☐ Data banks, data archives and/or medical records
☐ Sensitive Topics
☐ Non-English speaking subjects

The principal investigator assures the Committee that all procedures performed under the project will be conducted by individuals legally and responsibly entitled to do so and that any deviation from the project (e.g., change in principal investigatorship, subject recruitment procedures, drug dosage, research methodology, etc.) will be submitted to the review committee for approval prior to its implementation.

What do you plan to do with the results? Please provide a brief summary statement below:

I will use the results to complete my dissertation. I may additionally publish and present this content.
I hope to shine a light on the need to incorporate coding instruction in the Catholic school curriculum with the data that I will gather from schools in the Archdiocese of Los Angeles. I will share the results of this data with the participants and specifically the school where I am employed. I hope that the data will inform all our pedagogical decisions moving forward to best prepare students in a shifting global society.

Are you applying to a federal, state, foundation or any non-LMU organization for funding? If so, please list the source:

___N/A____

Please do not staple any documents submitted to the IRB. Please submit in single-sided form only. Double-sided documents will not be accepted.

NOTE: Applications and any additional material requested by the IRB will not be processed unless signed personally by the principal investigator.

Note: If your study involves recruiting participants from a school or other institution outside of LMU, please include a letter of permission, on official letterhead, signed by the head, director or principal of the institution. The letter should include the total number of participants, a brief explanation of the study and what study participants will be asked to do, and the amount of time human subjects will be expected to participate in the study. If your study involves a survey or questionnaire, the letter should indicate that the person providing permission has read and approved of those documents should be provided.

		Krikor K. Kiladjian
Date	Signature of Principal Investigator (Required)	Name (printed)
		Elizabeth C. Reilly
Date	Signature of Faculty Sponsor (Required)	Name (printed)
		Rebecca Stephenson
Date	Signature of Department Chair (Required)	Name (printed)

*Note: The department Chairperson's signature indicates they are aware of this research activity.
Electronic signature is acceptable.

Date | IRB Approval (Signature) | Name (printed)

IRB Approval Number

Please deliver to the IRB Office, University Hall Suite #1718

APPENDIX F

IRB Application Questionnaire

LOYOLA MARYMOUNT UNIVERSITY

IRB Application Questionnaire

All materials must be typed.

RESEARCH BACKGROUND
Please describe the purpose of your research. Provide relevant background information and briefly state your research question(s). You may provide relevant citations as necessary. (300 Word Max.)

The aim of this research is to assess K–12 Archdiocese of Los Angeles Catholic School educators', administrators', and STEM coordinators' understanding of coding, gauge their perceptions on coding's impact on student achievement in the classroom, occupational preparedness, and potential to diversify the STEM workforce, and evaluate whether their epistemology and pedagogy impacts their inclination to integrate constructionism and coding instruction.

There is increased prominence accorded to STEM, including computer science and in particular coding instruction, by educational and legislative leaders. Additionally, there is a growing sentiment amongst the American public that computer science and coding instruction are essential for future success. Consequently, I would like to amplify this momentum to ensure the fruition of a uniform shift in curriculum that embraces coding in Catholic schools in the ADLA.

My research questions are the following:

1. What is K–12 ADLA Catholic school teachers', administrators', and STEM directors'/technology coordinators' understanding of coding?
2. What is their perception on its instruction's impact on students' academic achievement, occupational preparedness, and on its ability to enhance STEM representation in school and in the workforce?
3. How do K–12 ADLA Catholic teachers', administrators', and STEM directors'/coordinators' epistemology and pedagogy impact their inclination to integrate the tenets of constructionism and coding?

SUBJECT RECRUITMENT
How will subjects be selected? What is the sex and age range of the subjects? Approximately how many subjects will be studied? How will subjects be contacted? Who will make initial contact with subjects? Specifically, what will subjects be told in initial contact? If subjects will be screened, describe criteria and procedures.

I focused on grade schools and high schools in the Archdiocese of Los Angeles, which is the largest diocese in the United States. The gender and age range will vary; however, my subjects will only be adults over the age of 18. It is my hope that I will receive responses from at least half of the educators and administrators in the over 200 grade schools and high schools in the ADLA.

The principal investigator contacted the assistant superintendent of the Department of Catholic Schools, Dr. Tony Galla, and received approval to conduct the study. The principal investigator also received approval to conduct the study in the high schools by former assistant superintendent of DCS high schools, Dr. Dan O'Connell. Dr. Galla sent me the list of principal emails to be strictly used for the purposes of this dissertation's research. Upon approval by the IRB, the principal investigator will send the letter to the principals requesting that their respective school take part in the survey.

PROCEDURES
Summarize fully all procedures to be conducted with human subjects.

The principal investigator contacted the assistant superintendents (grade school and high school) of the Department of Catholic Schools in the Archdiocese of Los Angeles and forwarded an official request to conduct the study. The letter explained the purpose and significance of the study. The requests were approved and the principal investigator received the list of principal emails, which he will use to forward a letter explaining the purpose of the study and requesting participation in the survey. At that point, the email recipients will have the opportunity to take part in the survey through Qualtrics. Those who complete the survey will be included in a raffle to potentially win one of five $25 dollar Amazon gift cards.

*Note—before the survey was sent to the potential participants, I requested six people to take the survey to identify issues. The testers provided invaluable feedback and with the guidance of Dr. Reilly, I integrated the feedback. Feedback ranged from length of survey to ambiguous language in a few questions.

RISKS / BENEFITS
What are the potential benefits to subjects and/or to others? What are the reasonably foreseeable risks to the subjects? (Risks may include discomfort, embarrassment, nervousness, invasion of privacy, etc.) If there are potential risks to subjects, how will they be minimized in advance? How will problems be handled if they occur?

The benefits of the study will highlight a need in Catholic schools across the ADLA. Coding instruction has the potential to enhance student achievement, better prepare our students for forthcoming occupations in a global society and serve to bridge the divide that exists in access to technology and academic and career opportunities for traditionally marginalized groups. The data from this study will ideally inform pedagogical decisions of school personnel in the ADLA to meet these ends.

Some school personnel may feel uncomfortable in acknowledging that they are not prepared or may not want to integrate coding instruction in the classroom. Otherwise, I do not anticipate any potential discomfort, embarrassment, invasion of privacy, or any other adverse effect from participating in the study.

CONFIDENTIALITY
Will subjects be identifiable by name or other means? If subjects will be identifiable, explain the procedures that will be used for collecting, processing, and storing data. Who will have access to data? What will be done with the data when the study is completed? If you are collecting visual images of your subjects please justify this.

Since this is a survey, the study will be both confidential and anonymous. Although the list of principal emails will identify the names of some potential respondents and their schools, each participant will be given an identifying number by Qualtrics. Therefore, subjects will not be identified by name in the study. Nevertheless, participants will be asked to provide demographic information including, but not limited to, their age, gender, level of education, role in the school, and subjects taught. All this information will be stored on a password-protected laptop with security software manufactured by Norton, which protects against viruses and other types of malware. All the data will be received through the principal investigator's LMU email, which is encrypted.

The data will then be entered in Qualtrics, which has its own security features. Furthermore, the laptop will be located in the principal investigator's home at all times, unless he takes it with him to work. Even in that case, the laptop would still remain within his control at all times. Only the principal investigator will have access to the data. The data will remain with the principal investigator even after the conclusion of the study for potential use in future studies. Visual images of the subjects will not be collected.

INFORMED CONSENT
Attach an informed consent form or a written request for waiver of an informed consent form. Include waiver of written consent if appropriate. If your research is being conducted in another language, please include copies of the translated "Informed Consent" or "Waiver of Written Consent" forms.

See Appendix G

STUDENT RESEARCH
When a student acts as principal investigator, a faculty sponsor signature is required on the application form.

Please see the cover page for my advisor's signature (see Appendix E).

RENEWAL APPLICATIONS
When the submission is a Renewal Application, include a summary of the research activities during the previous granting period specifically addressing: number of subjects studied and any adverse reactions encountered, benefits which have been derived, any difficulty in obtaining subjects or in obtaining informed consent, and approximate number of subjects required to complete the study.

N/A

PAYMENTS
If subjects are to be paid in cash, services, or benefits, include the specific amount, degree, and basis of remuneration. Participants will be entered into a raffle to win one of five $25 Amazon gift cards.

PSYCHOLOGY SUBJECT POOL
When students from the Psychology Subject Pool (PSP) are to be involved as subjects, permission must be obtained from the PSP prior to running subjects. Forms are available from the Psychology Office in 4700 University Hall. It is not necessary to inform the IRB of approval from the PSP, however the PSP requires IRB approval prior to permission for using the pool being granted.

N/A

QUALIFICATIONS AND TRAINING
Describe the qualifications of, or method of training and supervision afforded student experimenters. This includes past experience, type and frequency of student/sponsor interactions during the experiment, and Human Subjects Protections Training.

I have taken 6 methods courses during LMU's doctoral program. Furthermore, I have been under Dr. Elizabeth C. Reilly's supervision during the dissertation writing process and have been assisted by committee members Dr. Philip Molebash and Dr. Peter Rich.

Furthermore, I have completed the Collaborative Institutional Training Initiative program (CITI); see Appendix C for the certificate (p. 128).

RANDOMIZATION
Describe criteria for assigning subjects to sub-groups such as “control” and “experimental.”

N/A

USE OF DECEPTION
If the project involves deception, describe the debriefing procedures that will be used. Include, verbatim, the following statement in the consent form: “Some of the information with which I will be provided may be ambiguous or inaccurate. The investigator will, however, inform me of any inaccuracies following my participation in this study.”

N/A

QUESTIONNAIRES AND SURVEYS
Include copies of questionnaires or survey instruments with the application (draft form is acceptable). If not yet developed, please so indicate and provide the Committee with an outline of the general topics that will be covered. Also, when the questionnaire or interview schedule has been compiled, it must be submitted to the Committee for separate review and approval. These instruments must be submitted for approval prior to their use. Consider your population. If they are foreign speakers, please include copies in the foreign language.

See survey in Appendix K.

PHYSICIAN INTERACTIONS
To ensure that all patients receive coordinated care, the principal investigator is obligated to inform the primary physician (when not the principal investigator) of all studies on his/her patients.

N/A

SUBJECT SAFETY

Describe provisions, if appropriate, to monitor the research data collected, to ensure continued safety to subjects.

See Appendix I—Qualtrics Security Statement
See Appendix J—Norton

The same digital security apparatus mentioned above will be utilized to ensure the safety of the data (password protection, Norton security software, and Qualtrics’ security system. Furthermore, the laptop will primarily be kept at the principal investigator’s home.

REDUNDANCY
To minimize risks to subjects, whenever appropriate, use procedures already being performed on the subjects for diagnostic or treatment purposes. Describe provisions.

N/A

COUNSELING
In projects dealing with sensitive topics (e.g., depression, abortion, intimate relationships) appropriate follow-up counseling services must be made available to which subjects might be referred.
The IRB should be notified of these services and how they will be made available to subjects.

N/A

SAFEGUARDING IDENTITY
When a research project involves the study of behaviors that are considered criminal or socially deviant (i.e., alcohol or drug use) special care should be taken to protect the identities of participating subjects. In certain instances, principal investigators may apply for "Confidentiality Certificates" from the Department of Health and Human Services or for "Grants of Confidentiality" from the Department of Justice.

N/A

ADVERTISEMENTS
If advertisements for subjects are to be used, attach a copy and identify the medium of display.

See Appendix A for email requesting consent to conduct study in the Archdiocese of Los Angeles. See Appendix B for email soliciting schools in the Archdiocese of Los Angeles to participate in the survey.

FOREIGN RESEARCH
When research takes place in a foreign culture, the investigator must consider the ethical principles of that culture in addition to the principles listed above.

N/A

EXEMPTION CATEGORIES (45 CFR 46.101(b) 1-6)
If you believe your study falls into any of the Exemption Categories listed below, please explain which category(ies) you believe it falls into and why.

Research conducted in established or commonly accepted educational settings, involving normal educational practices, such as (i) research on regular and special instructional strategies, or (ii) research on the effectiveness of or the comparison among instructional techniques, curricula, or classroom management methods.
Research involving the use of educational tests (cognitive, diagnostic, aptitude, achievement), if information taken from these sources is recorded in such a manner that subjects cannot be identified, directly or through identifiers linked to the subjects.

Research involving survey or interview procedures, except where all the following conditions exist: (i) responses are recorded in such a manner that the human subjects can be identified, directly or through identifiers linked to the subjects, (ii) the subject's responses, if they became known outside the research, could reasonably place the subject at risk of criminal or civil liability, or be damaging to the subject's financial standing, employability, or reputation, and (iii) the research deals with sensitive aspects of the subject's own behavior, such as illegal conduct, drug use, sexual behavior, or use of alcohol. All research involving survey or interview procedures is exempt, without exception, when the respondents are elected or appointed public officials, or candidates for public office.

Research involving the observation (including observation by participants) of public behavior, except where all the following conditions exist: (i) observations are recorded in such a manner that the human subjects can be identified, directly or through the identifiers linked to the subjects, (ii) the observations recorded about the individual, if they became known outside the research, could reasonably place the subject at risk of criminal or civil liability, or be damaging to the subject's financial standing, employability, or reputation, and (iii) the research deals with sensitive aspects of the subject's own behavior such as illegal conduct, drug use, sexual behavior, or use of alcohol.

Research involving the collection or study of existing data, documents, records, pathological specimens, or diagnostic specimens, if these sources are publicly available or if the information is recorded by the investigator in such a manner that subjects cannot be identified, directly or through identifiers linked to the subjects.
Unless specifically required by statute (and except to the extent specified in paragraph (1)), research and demonstration projects which are conducted by or subject to the approval of the Department of Health and Human Services, and which are designed to study, evaluate, or otherwise examine: (i) programs under the Social Security Act or other public benefit or service programs, (ii) procedures for obtaining benefits or services under those programs, (iii) possible changes in or alternatives to those programs or procedures, or (iv) possible changes in methods or levels of payment for benefits or services under those programs.

Please deliver to: Julie Paterson, IRB Coordinator, University Hall, Suite 1718 or jpaterso@lmu.edu.

APPENDIX G

Informed Consent Form

Loyola Marymount University
Informed Consent Form

TITLE:	Perceptions of Coding Instruction in K–12 Catholic Schools in the Archdiocese of Los Angeles
INVESTIGATOR:	Krikor Kiladjian, School of Education, Loyola Marymount University, 818-307-3339
ADVISOR: (if applicable)	Elizabeth Reilly, School of Education, Loyola Marymount University, (310) 258-8803

PURPOSE
You are being asked to participate in a research project that seeks to investigate teachers', principals', and administrators' perceptions of coding instruction in K–12 Catholic schools in the ADLA. You will be asked to complete a survey that has 8 blocks and that will take approximately 15–20 minutes to complete.

RISKS
Risks associated with this study include: Some personal information will be asked including age, level of education, religious affiliation, and your perceived ability to integrate coding instruction in the classroom.

BENEFITS
I hope to gather data from educators, administrators, and STEM directors/technology coordinators from the Archdiocese of Los Angeles to demonstrate the need to integrate coding instruction in the ADLA.

INCENTIVES
All participants who complete the survey will be entered in a raffle to win one of five $25 Amazon gift cards. Participation in the project will require no monetary cost to you.

CONFIDENTIALITY
Participants names will not be collected, however, demographic data such as age, name of diocese in which the school is located, gender, subjects taught, and perceived ability to integrate coding instruction in schools will be requested. Your name will never be used in any public dissemination of these data (publications, presentations, etc.). All research materials and consent forms will be stored on the principal investigator's personal laptop, which Is password protected and is secured with an anti-virus/internet security software called Norton. The laptop will be kept primarily in the principal investigator's home, but it may additionally be taken to his workplace. Nevertheless, the laptop will always be under the watchful eye of the principal investigator. Qualtrics, which has its own digital security apparatus, will be utilized as the survey software. Even when the research study ends, the principal investigator will keep all data on his personal

laptop for potential use in future studies. All the information you provide will be kept confidential.

RIGHT TO WITHDRAW
Your participation in this study is *voluntary*. You may withdraw your consent to participate at any time without penalty. Your withdrawal will not influence any other services to which you may be otherwise entitled, your standing, or relationship with Loyola Marymount University.

SUMMARY OF RESULTS
A summary of the results of this research will be supplied to you, at no cost, upon request. The principal investigator's contact information is as follows: mobile: 818-307-3339; email: kkiladji@lion.lmu.edu. The summary will be made available approximately 4 weeks after the conclusion of the study.

VOLUNTARY CONSENT
I have read the above statements and understand what is being asked of me. I also understand that my participation is voluntary and that I am free to withdraw my consent at any time, for any reason, without penalty. If the study design or use of the information is changed I will be informed and my consent reobtained. On these terms, I certify that I am willing to participate in this research project.

I understand that if I have any further questions, comments or concerns about the study or the informed consent process, I may contact Dr. David Moffet, Chair, Institutional Review Board, Loyola Marymount University, 1 LMU Drive, Los Angeles, CA 90045-2659 or by email at David.Moffet@lmu.edu.

____ ____________________

Participant's Signature Date

APPENDIX H

LOYOLA MARYMOUNT UNIVERSITY

Experimental Subjects Bill of Rights

Pursuant to California Health and Safety Code §24172, I understand that I have the following rights as a participant in a research study:

1. I will be informed of the nature and purpose of the experiment.
2. I will be given an explanation of the procedures to be followed in the medical experiment, and any drug or device to be utilized.
3. I will be given a description of any attendant discomforts and risks to be reasonably expected from the study.
4. I will be given an explanation of any benefits to be expected from the study, if applicable.
5. I will be given a disclosure of any appropriate alternative procedures, drugs or devices that might be advantageous and their relative risks and benefits.
6. I will be informed of the avenues of medical treatment, if any, available after the study is completed if complications should arise.
7. I will be given an opportunity to ask any questions concerning the study or the procedures involved.
8. I will be instructed that consent to participate in the research study may be withdrawn at any time and that I may discontinue participation in the study without prejudice to me.
9. I will be given a copy of the signed and dated written consent form.
10. I will be given the opportunity to decide to consent or not to consent to the study without the intervention of any element of force, fraud, deceit, duress, coercion, or undue influence on my decision.

APPENDIX I

Qualtrics Security Statement

Qualtrics Security Statement

Security Statement

January 31, 2020

OUR SECURITY, BRIEFLY STATED

Qualtrics' most important concern is the protection and reliability of customer data. Our servers are protected by high-end firewall systems and scans are performed regularly to ensure that any vulnerabilities are quickly found and patched. Application penetration tests are performed annually by an independent third-party. All services have quick failover points and redundant hardware, with backups performed daily.

Access to systems is restricted to specific individuals who have a need-to-know such information and who are bound by confidentiality obligations. Access is monitored and audited for compliance.

Qualtrics uses Transport Layer Security (TLS) encryption (also known as HTTPS) for all transmitted data. Surveys may be protected with passwords. Our services are hosted by trusted data centers that are independently audited using the industry standard SSAE-18 method.

ISO 27001 Certification

In April 2018, Qualtrics achieved ISO 27001 certification. The direct link to the information and certificate is: https://www.schellman.com/certificate-directory?certificateNumber=1723268-3. To independently verify the status of the certification, please visit https://www.schellman.com/certificate-directory.

schellman

CERTIFICATE OF REGISTRATION

Information Security Management System - ISO/IEC 27001:2013

The Certification Body of Schellman & Company, LLC hereby certifies that the following organization operates an Information Security Management System that conforms to the requirements of ISO/IEC 27001:2013

Qualtrics, LLC

FedRAMP Authorization

Qualtrics is FedRamp Authorized. FedRAMP is the standard of U.S. government security compliance, with over 300 controls based on the highly-regarded NIST 800-53 that requires constant monitoring and periodic independent assessments. More information is found at https://www.fedramp.gov.

HITRUST

To better support our healthcare customers, Qualtrics achieved the HITRUST certification in September 2018. The validated report is available upon request to your account executive.

More Information

Qualtrics customers may request various security-related documents and questionnaires by contacting their account executive.

APPENDIX J

Norton Privacy Statement

Introduction

When it comes to your personal data, NortonLifeLock (2022) and its subsidiaries (collectively referred to as "NortonLifeLock", "Norton", "LifeLock", "we", or "us"), as well as our employees, contractors, and service providers, are committed to providing you with transparency. We process personal data in accordance with applicable legislation.

We do not, and will not without consent, sell, lease, or rent your information to third parties for monetary or other valuable consideration.

This Privacy Statement ("Statement") applies to the NortonLifeLock websites, services, and products (our "Services") that link to or reference this Statement. In this Statement, we describe how we collect, process, and share personal data, and the choices available to you regarding collection and use of your personal data.

Additional information on our personal data practices may be provided in product descriptions, contractual terms, supplemental privacy statements, or notices provided prior to or at the time we collect your data. Please see www.nortonlifelock.com/privacy for more details about what personal data we collect and how we process that personal data.

If you are in the European Economic Area, and unless stipulated otherwise contractually, the Controller of your personal data is:

NortonLifeLock Ireland Limited
Ballycoolin Business Park
Blanchardstown
Dublin 15
Ireland D15E867

Categories of Personal Data We Collect
The categories of personal data that we collect, and the general purposes of collection, include:

- User Data. This data is necessary to establish your account and method/means of payment, ship and provide your services, and track and maintain our relationship with you. It includes: o Name, mailing address, email address, phone number, and user credentials (login name and password); o Shipping and billing data, including credit card and payment data;

    o Your social security number and/or state/government identifier, driver's license number or other government-issued identifier, date of birth, age, gender, and other personal details about you as necessary to verify your identity and to provide identity theft protection services; o Bank transaction and other alert data depending on the product and features you have selected; and o Your transaction and support history with us.

- Administrative Data. This includes data we receive from you and from third parties that is necessary to facilitate installation of our products and may include device and system information, product license information, and/or usage and preference information, including how you would like us to communicate with you. This data helps us better understand and better serve your interests, expectations, needs, and requirements.

- Security Data. This data may include financial transactions, location data, and data that is collected for cyber threat intelligence, as needed to provide cyber safety and identity threat protection Services.

- Diagnostic Information. This data may include application crash reports and information you provide to us for obtaining customer service, as necessary, to troubleshoot any malfunctioning Services. For example, when you call or exchange email, chat, or mail with us, we may retain and review call and chat recordings and/or the contents of the messages as required/permitted by law and our recording and information management policies.

- Third-Party Data. This information includes data we may obtain about you from a third party, or data provided about you from a third party through the use of our Services, including threat intelligence data used to analyze threats and protect you, us, and our other customers against cyber threats. Third-party data may include the email and IP address of the sender of malicious malware.

The Purpose of Processing Your Personal Data
We collect and process your personal data for the following reasons:

For the purpose of fulfilling our contract with you at your request, in order to:
- Create and manage your account;
- Authenticate your identity prior to enrolling in our Services;
- Verify your identity and entitlement to Services, when you contact us or access our Services;
- Process your purchase transactions;
- Update you on the status of your orders;
- Allow you to register the Services you purchase;
- Manage your subscriptions; and
- Provide you with technical and customer support.

For specific purposes based on your consent, in order to:
- Subscribe you to newsletters and send you product updates or technical alerts;
- Send you marketing communications and information on new Services;
- Communicate with you about, and manage, your participation in contests, offers, or promotions;
- Solicit your opinion or feedback and/or provide opportunities for you to test Services;
- Enable you to refer a friend who may be interested in our offerings, as permitted by law;
- As applicable, to enable non-essential cookies or similar technologies;
- Process sensitive or special category personal data in accordance with applicable law; and
- As applicable, to provide you with interest-based ads about NortonLifeLock on sites other than our own.

For the purpose of fulfilling our legal obligations, we may be obligated to, for instance, keep and process records for tax purposes, accounting, other obligations such as court or other legal orders, and other necessary disclosures.

For the purpose of promoting and operating our business and advancing our legitimate interests, such as the effective delivery of our Services, and communications to you as well as to our other customers and partners, in order to:

- Provide you with information and Services that you request;
- Enable participation in interactive features of our Services, and notify you about changes to our terms or this Privacy Statement;
- Communicate commercial promotions and provide quotes for our Services;
- Promote and administer co-branded offers with trusted partners;
- Confirm sales conversions and conduct lead generation activities;
- Better administer and understand the usability, performance, and effectiveness of our Services, including troubleshooting, data analytics, testing, research, and statistical analysis;
- Improve our Services (including developing new Services) and customize and present content in the most relevant and effective manner for you and for your device, including suggestions and recommendations about things that may be of interest to you;
- *Enhance the security of our own networks and information systems;
- *Develop cyber-threat intelligence resources; and
- *Otherwise keep our Services, business, and users safe and secure, and comply with applicable laws and regulations to protect or exercise our legal rights and defend against legal claims.

<u>*For Network and Information Security Purposes and Cyber-Threat Intelligence</u>:

Our legitimate interests include developing threat intelligence resources aimed at maintaining and improving the ability of our information networks and systems to resist unlawful or malicious actions and other harmful events, such as cybercriminal activities, and attempts at identity theft or fraud ("cyber and identity threats").

The personal data we process for network and information security purposes includes, without limitation, network traffic data related to cyber and identity threats such as:

- Sender email addresses (e.g., of sources of SPAM such as phishing scams);
- Recipient email addresses (e.g., of victims of targeted email attacks);
- Reply-to email addresses (e.g., as configured by cybercriminals sending malicious email);
- Filenames and execution paths (e.g., of malicious or otherwise harmful executable files attached to emails);
- URLs and associated page titles (e.g., of web pages broadcasting or hosting malicious or otherwise harmful content);
- IP addresses (e.g., of web servers and connected devices involved in the generation, distribution, conveyance, hosting, caching, or other storage of cyber and identity threats such as malicious or otherwise harmful content); and/or
- Browser information (e.g., user agent string and session within cookies).

Depending on the context in which such data is collected, the data may contain personal data concerning you or third parties. However, in such cases, we will process the data only to the extent strictly necessary and proportionate to the purposes of detecting, blocking, reporting (by removing any personally identifiable elements), and mitigating the cyber or identity threats of concern or those of other users relying on our Services to protect their networks, systems, and identities. When processing personal data in this context, we will only identify specific data subjects if and to the extent necessary for the remediation of the cyber or identity threats concerned, or as required by law.

Please be aware that if it is determined that personal data concerning you is processed by NortonLifeLock because it is necessary for the detection, blocking, or mitigation of convicted cyber or identity threats, then objection, rectification, or erasure requests may be rejected in accordance with applicable law.

How We Collect Your Personal Data
We collect personal data about you from the following sources:

Data you provide:
- When you interact directly with us, we may collect personal data that you provide to us (e.g., account and payment information).

Data we collect automatically:
- When you visit and use our websites and Services, we may automatically collect data about your interaction with our websites and Services.

Data we collect about you from third parties:
- Credit reporting agencies and financial institutions (used for purposes such as identity theft protection Services);
- Marketing and joint-marketing partners (used for purposes such as to offer Services and/or joint Service bundles to prospective members);
- Public sources such as the dark web to alert you to potential misuse of your data;
- Private sources for purposes of providing customers with alerts related to financial transactions, property title, social media abuse, and other types of alerts within our products; and
- We may collect personal data from you about other people, such as personal data about friends and family through customer or employee referrals, or data about family members you include on your account.

When and Why We Share Your Personal Data
We are committed to maintaining your trust, and we want you to understand when and with whom we may share the information we collect. We do not, and will not without consent, sell, lease, or rent your information to third parties for monetary or other valuable consideration.

We permit third parties to process information necessary for our Services, and we may disclose the personal information we process to third parties for business purposes. The third parties we work with are contractually required to comply with adequate privacy, confidentiality, and security requirements.

With Our Partners
We may provide your user data, administrative data, security data, and third-party data to our partners for the purpose of allowing them to conduct NortonLifeLock business. Our partners may use your personal data to communicate with you and others about NortonLifeLock Services either alone or jointly with partner products and services. We may provide your personal data to partners to confirm your eligibility for joint or co-branded offers or to communicate and administer such offers (e.g., report sales conversions, verify eligibility, assess effectiveness of joint offer, etc.). Our partners are not allowed to use personal data that they receive from us for any purpose except for communicating, evaluating, improving, and administering the offer in question (NortonLifeLock branded, co-branded, or joint offer). This will not affect the partner's ability to use personal data that it may already have obtained from you or other sources. If you do not wish to receive promotional emails from our partners, you can unsubscribe directly using the unsubscribe link or tool provided in the partner's email or other communication to you.

With Service Providers Processing Data on Our Behalf
We may use contractors and service providers to process the personal data we collect for the purposes described in this Statement, the relevant Product and Service Privacy Statements, and for business purposes such as financial auditing, data storage and security, troubleshooting and debugging, improving and operationalizing threat intelligence and counter-threat measures, and for marketing and promoting our Services.

We contractually require service providers to keep data confidential, and we do not allow our service providers to disclose your personal data to others without our authorization, or to sell it or use it for purposes unrelated to the services they provide (e.g., their own marketing purposes). However, if you have a separate and/or independent relationship with these service providers, their privacy statements will apply to such relationships. Such service providers may include benefit brokers, your employer (for products and services offered as an employee benefit), contact centers, payment card processors, and marketing, survey, or analytics suppliers.

With Public Authorities and Legal Proceedings
In certain instances, it may be necessary for us to disclose any of the personal data we collect to comply with a legal obligation, at the request of public authorities, or as otherwise required by applicable law. No personal data will be disclosed except in response to:

- A subpoena, warrant, or other legal process issued by a court or other public authority of competent jurisdiction;
- Discovery requests or demands as part of a civil lawsuit or similar legal process;
- Where disclosure is necessary for us to enforce our legal rights pursuant to applicable law;
- A request with the purpose of identifying and/or preventing credit card fraud or identity theft; or
- Where disclosure of personal data is necessary to prevent or lessen a serious and imminent threat of bodily or other significant harm to the data subject or other individuals potentially concerned.

For Restoration Services
We may disclose your user data, security data, diagnostic information, and third-party data to financial institutions, financial services companies, and other third parties at your direction to provide restoration services and other Services to you.

With Third-Party Service Providers
If you access third-party services through our Services, these third-party services may be able to collect user data, security data, diagnostic information, and third-party data about you in accordance with their own privacy policies.

With Our Corporate Affiliates
We may share all the information we collect with our corporate affiliates.

For Business Transfers
We may share all the information we collect in connection with a substantial corporate transaction, such as the sale of a website, a merger, acquisition, consolidation, asset sale, or initial public offering, or in the unlikely event of bankruptcy.

Retention and Deletion of Your Personal Data
We will keep your personal data on our systems as long as necessary to provide you with our Services, or for as long as we have another legitimate business purpose to do so, but not longer than permitted or required by law. When we no longer have an ongoing legitimate business reason to keep your personal data, your personal data will either be securely disposed of, or de-identified through an appropriate

anonymization means, such as aggregation, truncation, or one-way hashing so it is no longer identifiable as your personal data.

Cross-Border Transfers of Personal Data Among NortonLifeLock Entities and to Third-Party Vendors

We are a global company and process personal data in many countries. As part of our business, your personal data may be transferred to NortonLifeLock and/or its subsidiaries and affiliates in the United States, and to subsidiaries and third-party vendors of NortonLifeLock located worldwide. All transfers will occur in compliance with the applicable data transfer requirements laws and regulations.

If your personal data originates from the European Economic Area and is transferred to NortonLifeLock subsidiaries, affiliates, or third-party vendors engaged by NortonLifeLock to process such personal data on our behalf who are located in countries that are not recognized by the European Commission as offering an adequate level of personal data protection, such transfers are covered by alternate appropriate safeguards, specifically Standard Contractual Clauses adopted by the European Commission.

If we are involved in a reorganization, merger, acquisition, or sale of our assets, your personal data may be transferred as part of that transaction.

How We Protect Your Personal Data

Securing personal data is an important aspect of protecting privacy. We take reasonable and appropriate physical, technical, and organizational security measures in accordance with applicable laws to protect your personal data against the risk of accidental loss, compromise, or any form of unauthorized access, disclosure, or processing. The relevant security controls are communicated throughout NortonLifeLock to support the secure development of Services and maintain a secure operating environment. Our security approach includes:

Physical Safeguards

We lock doors and file cabinets, control access to our facilities, implement a clean desk policy, and apply secure destruction to media containing personal data.

Technical Safeguards

We implement and use reasonably available state-of-the-art network and information security standards, protocols, and technologies, including encryption, intrusion detection, and data loss prevention, and we monitor our systems and data centers to comply with our security policies.

Organizational Safeguards

We conduct regular company-wide as well as role-specific training and awareness programs on security and privacy.

If you have any questions about the security of your personal data or the security of the site, or wish to report a potential security issue, please contact security@nortonlifelock.com. When reporting a potential security issue, please describe the matter in as much detail as possible and include any information that might be helpful. If you are having problems accessing your account, please contact our Member Support Center.

Your Privacy Rights

You can view and update your personal data through your Norton Account or LifeLock Portal. There are a variety of data protection laws around the globe that provide privacy rights to you as our customer. Subject to applicable laws, you may have the following rights:

- Delete: Right to delete or erasure ("right to be forgotten") of personal data we have collected about you;
- Access: Right to access the personal data we have collected about you, as well as other information about our data processing practices;
- Rectify: Right to rectify, correct, update, or complement inaccurate or incomplete personal data we have about you;
- Restrict: Right to restrict the way we process your personal data;
- Withdraw Consent: Right to withdraw your consent to process your personal data;
- Object: Right to object to our processing of your personal data based on legitimate interest;
- Portability of Personal Data: Right to obtain a portable copy of your personal data; and
- Lodge a Complaint: Right to lodge a complaint with a supervisory authority if you are not satisfied with the way we have handled your personal data, or any privacy request, or other request that you have raised with us.

To exercise any of your rights, or to raise any other questions, concerns, or complaints about our privacy practices, or about our use of your personal data and its privacy, or if you are not a customer of ours and want to know what personal data we have about you, please contact us as explained below ("Contact Us").

Once we receive your request, we will verify your identity and your authorization to take the actions requested, authenticating your identity at a level appropriate to the requested action. We require you to re-authenticate before we will disclose or delete data. You may be entitled, in accordance with applicable law, to submit a request through an authorized agent. To designate an authorized agent to exercise your rights and choices on your behalf, please contact NortonLifeLock Support.

In some instances, we are unable to delete or erase your personal data upon request as a result of an ongoing legal obligation such as a legal hold or court order. Depending on your location, there may be other restrictions or exemptions to full deletion or erasure of your personal data.

We will not discriminate against you for exercising your rights and choices, although some of the functionality and features available on a Service may change or no longer be available to you where the processing of certain data is essential to the use of the Service or feature.

Contact Us

Americas
NortonLifeLock Inc.—Privacy
60 East Rio Salado Parkway, Suite 1000
Tempe, AZ 85281
Email: nll_privacy@nortonlifelock.com
Member Services: 1-800-543-3562

Europe—Middle East—Africa
NortonLifeLock Ireland Limited—Global Privacy Office
Ballycoolin Business Park
Blanchardstown
Dublin 15
Ireland D15E867
Email: nll_privacy@nortonlifelock.com

Independent EU GDPR Data Protection Officer
Pembroke Privacy Ltd
4 Upper Pembroke Street
Dublin 2
Ireland DO2VN24
Email: DPO@nortonlifelock.com

Referrals
When you choose to provide us with personal data about third parties, we will only use this data for the specific stated reason that you provided it. It is your responsibility to abide by applicable privacy and data security laws when you disclose third parties' personal data to us, including informing third parties that you are providing their personal data to us and how it will be transferred, used, or processed, and securing the appropriate legal permissions and safeguards. If you choose to provide us with a third party's personal data, you represent that you have the third party's permission to do so. Examples include forwarding references or sending job referrals. You also acknowledge that when we interact with such third-party individuals whose personal data you share with us, it is our duty to inform them that we obtained their personal data from you. Where applicable, third parties may unsubscribe from any future communication following the link provided in the initial message. If you believe that one of your contacts has provided us with your personal data and you would like to request that it be removed from our database, please contact us.

Marketing and Community Networking
You may receive marketing messages and materials from us or our affiliates. You have choices on what communications you wish to receive from us. If you do not want to continue receiving any marketing materials from us, you have the following options:

- Click on the unsubscribe function in the communications you receive from us;
- Unsubscribe from Norton Marketing Offers;
- Manage your communication preferences in your Norton Account or LifeLock Portal;
- Contact our Member Services Department at 1-800-543-3562; or
- Contact our Member Services Department by regular mail at Attn.: Member Services, 60 East Rio Salado Parkway, Suite 1000, Tempe, AZ 85281.

If you choose not to receive marketing communications from us, we will honor your request.

However, we will continue to communicate with you as needed to provide the Services you are entitled to, to respond to your inquiries, or to otherwise relay transactional product or service-related messages.

Please also be aware that you may still receive information about our Services through other parties using their own mailing lists. For instance, marketing materials for our Services may also be contained in messages you receive from third parties, such as your employer if they offer our Services as part of their employee benefits.

Interest-Based Ads
We may provide your data, including the data about your interests in our Services, to third parties for the purposes of serving you more relevant ads about our Services. Where we provide you with interest-based ads on a site other than our own, we do not track your other activities on that site. If you click on our ads, we will know the domain you came from. For more information, please see the Cookies & Tracking Technologies section below.

Data From Third Parties

Third parties may provide us with personal data they have collected about you, from you, or from other online and offline sources. Marketing data from our partners and third parties can be combined with information we already have about you, to provide you with more relevant communications and to better tailor our offers to you. We make reasonable efforts to verify that the third parties we work with for marketing purposes are reputable, and we do not ask them to disclose your personal data if we do not have a lawful purpose and valid legal basis to collect and process that data.

Community Networking—NortonLifeLock Community (Forums, Blogs, and Networking Sites)
We operate forums, websites, and related information services, to better assist you in using our Services, discussing technical issues, and sharing your experiences. You should be aware that any data you provide in these public forums will be read, collected, and used by others who access them. To request removal of your personal data from any forum, contact us here. In certain circumstances, we may not be able to remove your personal data, in which case we will let you know why. Please note that your use of these community services may be subject to additional terms and conditions.

How to Opt Out of Interest-Based Advertising
We partner with third parties to display advertising on our website or to manage our advertising on other sites. You may opt out of many third-party ad networks, including those operated by members of the Network Advertising Initiative (NAI) and the Digital Advertising Alliance (DAA). For more information and available choices for third-party ad networks participating in the NAI and DAA programs, please visit their respective websites: www.networkadvertising.org/optout_nonppii.asp (NAI) and www.aboutads.info/choices (DAA). If you are in the European Union, you may also do so by visiting Your Online Choices (click here). Please note that if you opt out, you will continue to receive generic ads not based on your interests. Opting out of these networks does not otherwise limit the collection of information described elsewhere in this Statement.

Note: If your browser is configured to reject cookies when you visit the opt-out page, or you subsequently erase your cookies, use a different computer, or change web browsers, your opt-out may no longer be effective.

Cookies & Tracking Technologies

Cookies
A cookie is a commonly used automated data collection tool. Cookies are small text files that are placed on your computer or device by websites that you visit or HTML-formatted emails you open, to make websites work, or work more efficiently. We and our partners use cookies and other similar tracking technologies such as Web Beacons, JavaScript, clear GIFs, pixel tags, HTML5, and Flash Local Storage Objects on our websites or emails to:

- Enable the proper functioning of our websites and the proper delivery of legitimate electronic communications;
- Tailor information presented to you based on your browsing preferences, such as language and geographical region; • Collect statistics regarding your website usage;
- Provide us with business and marketing information; and
- In some cases, to enable a third party to deliver future advertising for our Services to you when you visit certain websites owned by third parties.

We use different kinds of cookies:

- Essential cookies are necessary to provide you with Services and features available through our websites. Without these cookies, services you may need, such as shopping carts or e-billing, cannot be provided.

- Advertising cookies and tracking scripts are used to make advertising messages more relevant to you. They perform functions like preventing the same ad from continuously reappearing, ensuring that ads are properly delivered and, in some cases, featuring ads based on your interests.
- Social Media cookies collect data regarding social media interactions.
- Analytics cookies collect data that is either used in aggregate form to help us understand how the website is being used or to understand the effectiveness of our marketing campaigns.
- Performance and Functionality cookies collect data that is used to help make the user experience better on our websites.

The cookies we use include "session" cookies that are erased when you leave our websites, or they may be "persistent" cookies that remain on your computer's hard drive after you leave the site, in preparation for your next visit to our websites.

If you do not wish to receive cookies, you may be able to refuse them by not agreeing to the use of them upon entering the website. If you do so, we may be unable to offer you some of our functionalities, Services, or support. If you have previously visited our websites, you may also have to delete any existing cookies from your browser. If you would like to view or manage the cookies we use, please see our cookie banner which is displayed when you first access our website, or the "privacy settings" tab located on the lower right hand of our websites.

Third-Party Data Collection
Cookies may also be placed on our websites by third parties to deliver tailored information and content that may be of interest to you, such as promotions or offerings, when you visit third-party websites after you have left ours. We do not permit these third parties to collect personal data about you on our sites beyond such cookies (e.g., email address) unless such data is provided to the third party in their role as a service provider acting solely on our behalf.

Social Media Features and Widgets
Our websites include social media features, such as Facebook's "Like" button, and widgets, such as the "Share This" button, or interactive mini-programs that run on our sites. These features are usually recognizable by their third-party brand or logo and may collect your IP address or other personal data, including which page you are visiting on our websites, and set a cookie to enable the feature to work properly. Any collection of information through such third-party feature is governed by the privacy policy of the company providing it.

Do-Not-Track Signals and Similar Mechanisms
Some mobile and web browsers transmit "do-not-track" signals. Because of differences in how web browsers incorporate and activate this feature, it is not always clear whether users intend for these signals to be transmitted, or whether they are even aware of them. We currently do not respond to these signals.

Automated Individual Decision Making and Profiling
Where NortonLifeLock processes network traffic data for the purpose of network and information security based on our or our customers' legitimate interest as outlined in the corresponding section of this Statement, automated decisions concerning data elements may occasionally be made. This could involve assigning relative cybersecurity reputation scores to IP addresses and URLs based on objective cyber-threat indicators measured by our and our partners' cyber-threat detection engines. Such indicators may be, for instance, the determination that malicious or otherwise harmful contents are hosted at a given URL or are coming from a given IP address. Such automatically assigned reputation scores may be leveraged by you, by NortonLifeLock, by our partners, and by other customers to detect, block, and mitigate the identified cyber threats. They could therefore result in our Services blocking network traffic coming from

or going to such URLs and IP addresses. This processing is intended only to protect you, NortonLifeLock, our partners, and our other customers from cyber threats. If you consider that such automated processing is unduly affecting you in a significant way, please contact us as explained above ("Contact Us") to raise your concerns and to seek our help in finding a satisfactory solution.

Children's Privacy
Our websites are not directed to, nor do we knowingly collect data from, minors (as defined by applicable law) except where explicitly described otherwise in the privacy notices of Services designed specifically for purposes such as to assist you by providing child online protection features. In such cases, we will only collect, and process personal data related to any child under 13 years of age that you choose to disclose to us or otherwise instruct us to collect and process. Please refer to the Product Specific Privacy Statements for additional information.

Changes to This Statement
We reserve the right to revise or modify this Statement. In addition, we may update this Privacy Statement to reflect changes to our data practices. If we make any material changes in the way we collect, process, and/or share your personal data, we will notify you by email (sent to the e-mail address specified in your account) or by means of a notice on this website prior to the change becoming effective. In the case of a material change to our personal data processing practices, any such change will only apply on a going-forward basis. We will not process the personal data currently in our possession in a materially different way without your prior consent. We encourage you to periodically review this page for the latest information on our privacy practices.

Links to Other Websites
Our websites may contain links to other websites owned or operated by other companies. If you visit any linked websites, please review their privacy statements carefully. We are not responsible for the content or privacy practices of websites that are owned by those third parties. Our websites may also link to co-branded websites that are maintained by NortonLifeLock and one or more of our business partners who are collecting your personal data pursuant to their own privacy practices. Please review the applicable privacy statements on any co-branded site you visit, as they may differ from ours.

Disclaimer
This Privacy Statement does not apply to NortonLifeLock affiliates: (1) Avira Operations GmbH & Co. KG., including its related entities; and (2) BullGuard Limited, including its related entities. These entities maintain separate privacy statements which can be found on their respective websites.

EFFECTIVE DATE: March 1, 2021

APPENDIX K

Survey Instrumentation

Q1 Loyola Marymount University Informed Consent Form

TITLE:	Perceptions of Coding Instruction in K–12 Catholic Schools
INVESTIGATOR:	Krikor Kiladjian, School of Education, Loyola Marymount University, (818) 307-3339
ADVISOR:	(if applicable) Elizabeth Reilly, School of Education, Loyola Marymount University, (310) 258-8803

PURPOSE
You are being asked to participate in a research project that seeks to investigate teachers', principals', and administrators' perceptions of coding instruction in K–12 Catholic schools in the Archdiocese of Los Angeles. The survey will take approximately 15-20 minutes to complete.

RISKS
Risks associated with this study include: Some personal information will be asked including age, level of education, the location of the school's diocese, and your perceived ability to integrate coding instruction in the classroom.

BENEFITS
I hope to gather data from several different dioceses around the country to demonstrate the need to integrate coding instruction in Catholic schools.

INCENTIVES
All participants who complete the survey will be entered in a raffle to win one of 5 $20 Amazon gift cards. Participation in the project will require no monetary cost to you.

CONFIDENTIALITY
Participants' names will not be collected, however, demographic data such as age, name of diocese in which the school is located, gender, subjects taught, and perceived ability to integrate coding instruction in schools will be requested. Your name will never be used in any public dissemination of these data (publications, presentations, etc.). All research materials and consent forms will be stored on the principal investigator's personal laptop, which Is password protected and is secured with an anti-virus/internet security software called Norton. The laptop will be kept primarily in the principal investigator's home, but it may additionally be taken to his workplace. Nevertheless, the laptop will always be under the watchful eye of the principal investigator. Qualtrics, which has its own digital security apparatus, will be utilized as the survey software. Even when the research study ends, the principal investigator will keep all data on his personal laptop for potential use in future studies. All the information you provide will be kept confidential.

RIGHT TO WITHDRAW
Your participation in this study is *voluntary*. You may withdraw your consent to participate at any time without penalty. Your withdrawal will not influence any other services to which you may be otherwise entitled, your standing, or relationship with Loyola Marymount University.

SUMMARY OF RESULTS
A summary of the results of this research will be supplied to you, at no cost, upon request. The principal investigator's contact information is as follows: mobile: (XXX) XXX-XXXX; email: xxx@lion.lmu.edu. The summary will be made available approximately 4 weeks after the conclusion of the study.

VOLUNTARY CONSENT
I have read the above statements and understand what is being asked of me. I also understand that my participation is voluntary and that I am free to withdraw my consent at any time, for any reason, without penalty. If the study design or use of the information is changed, I will be informed and my consent reobtained. On these terms, I certify that I am willing to participate in this research project. I understand that if I have any further questions, comments or concerns about the study or the informed consent process, I may contact Dr. David Moffet, Chair, Institutional Review Board, Loyola Marymount University, 1 LMU Drive, Los Angeles, CA 90045-2659 or by email at David.Moffet@lmu.edu.

- o Yes, I do consent. (1)
- o No, I do not consent. (2)

Skip To: End of Survey If Loyola Marymount University Informed Consent Form TITLE: Perce... = No, I do not consent.

Q2 This survey will assess the perceptions of Archdiocese of Los Angeles educators, STEM directors/coordinators, and administrators regarding what coding instruction entails, its consequent benefits, means of implementation, and the parallel between epistemology and pedagogy. Teachers and administrators of all grade levels and content areas are invited to take part in this survey. The survey is composed of 72 questions, and it will take approximately 15–20 minutes to complete. One question at the conclusion of the survey asks respondents if they would like to submit their email address to be entered into a pool to win one of 5 Amazon gift cards. All the substantive questions are fixed choice in nature, meaning that there will be answer options for you to quickly choose from. Some questions will allow you to choose multiple answer choices. Furthermore, depending on your answer to certain questions, you will skip the very next question. Therefore, do not be surprised if a number skips. Additionally, question number 53 will prompt you to order your implemented teaching methods. Simply click and drag the answer choices to meet your desired order.

The survey is organized in the following eight blocks:
1. Informed Consent (1 question)
2. Instructions to Survey (1 question)
3. Understanding of Coding (18 questions)
4. Benefits of Coding (15 questions)
5. Implementation of Coding (17 questions)
6. Epistemology (9 questions)
7. Demographics (11 questions)
8. Raffle Participation (1 question)

Your responses are invaluable and I appreciate your time.

A definition of coding is important to provide context to this survey. Coding is the language that provides instructions to applications and software, thereby allowing them to function. Coding instruction takes place in the classroom as educators teach students how to code, whether it is embedded in other classes such as math or science or it serves as its own standalone course.

Q3 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
My students will want to learn coding. (1)	○	○	○	○	○

Q4 Why do you think your students will want to learn coding (you may choose more than one answer choice)?

- it is exciting/fun (1)
- it is something new (2)
- can collaborate with other students (3)
- other (please specify) (4) ____

Q5 Why do you *not* think your students will want to learn coding (you may choose more than one answer choice)?

- difficult (1)
- no prior experience (2)
- not interested (3)
- other (please specify) (4) ____

Q6 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
There is a disparity in access to technology in Catholic schools in the ADLA. (1)	○	○	○	○	○

Q7 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Girls are generally less involved in coding compared to boys in school. (1)	○	○	○	○	○

Display This Question:
If Please respond to the following statement. = Strongly agree
Or Please respond to the following statement. = Somewhat agree

Q8 Why do you strongly agree or somewhat agree that girls are less involved in coding?

- Less interest (1)
- Societal impact (perception that coding is more suitable for boys) (2)
- both a & b (3)
- Other (please specify) (4) ____

Q9 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
There are specific ethnic minorities that lack exposure to coding instruction in schools. (1)	o	o	o	o	o

Q10 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Students who intend to pursue a career related to programming/coding should be the only ones who are taught coding in schools. (1)	o	o	o	o	o

Q11 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
It is valuable to learn coding. (1)	o	o	o	o	o

Q12 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Male and female students can succeed in coding classes. (1)	o	o	o	o	o

Q13 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Boys are better at coding than girls. (1)	○	○	○	○	○

Q14 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Girls are better at coding than boys. (1)	○	○	○	○	○

Q15 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
There is a gap in male and female interest in STEM subjects in K–12 Catholic schools. (1)	○	○	○	○	○

Q16 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
There is a gap in male and female representation in STEM occupations (i.e, coding/computer programming, information technology. (1)	○	○	○	○	○

Q17 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
There is a gap in interest in K–12 STEM subjects between different ethnic groups. (1)	○	○	○	○	○

Q18 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
There is a divide in STEM occupation representation between different ethnic groups. (1)	o	o	o	o	o

Q19 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
I have a general understanding of what coding is. (1)	o	o	o	o	o

Q20 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
I have a more detailed understanding of what coding entails (examples: I understand how some coding apps work, I have used some coding apps myself, or I have taken coding courses in school) (1)	o	o	o	o	o

Q21 Why should coding be taught in schools (you may choose more than one answer choice)?

- o develops skills that improve student achievement (1)
- o better prepares students for future occupations (2)
- o Is a pathway to social justice by giving students who are usually marginalized a chance to code (3)
- o all of the above (4)
- o other (please specify) (5) ____

Q22 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding develops a student's logical thinking ability. (1)	o	o	o	o	o

Q23 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding develops a student’s problem-solving ability. (1)	○	○	○	○	○

Q24 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding develops a student’s creativity. (1)	○	○	○	○	○

Q25 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding develops a student’s collaboration skills. (1)	○	○	○	○	○

Q26 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding develops a student’s social skills. (1)	○	○	○	○	○

Q27 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Learning coding can help a student perform better in different subjects. (1)	○	○	○	○	○

Q28 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Learning coding helps a student be more engaged in school. (1)	○	○	○	○	○

Q29 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding is an important 21st century literacy. (1)	○	○	○	○	○

Q30 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Learning coding in K–12 will better prepare students to earn positions in technology-specific fields. (1)	○	○	○	○	○

Q31 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Learning coding in K–12 will better prepare students to earn positions in non-technology specific fields, such as medicine, retail, and government, (1)	○	○	○	○	○

Q32 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding can help bridge the gap in K–12 STEM interest between males and females. (1)	○	○	○	○	○

Q33 Please respond to the following question.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding can help bridge the gap in K–12 STEM interest between different ethnic groups. (1)	○	○	○	○	○

Q34 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Exposure to coding in K–12 schooling will help bridge the gap in STEM occupations held by males and females. (1)	o	o	o	o	o

Q35 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Exposure to coding in K–12 schooling will help bridge the gap in representation amongst different ethnic groups in STEM occupations. (1)	o	o	o	o	o

Q36 What challenges do you think teachers may face with coding instruction in the classroom (you may choose more than one answer choice)?

- Insufficient time (1)
- Lack of knowledge/preparation to teach (2)
- Lack of access to technology (i.e., devices, wi-fi network) (3)
- Lack of student interest (4)
- Other (please specify) (5) ____

Q37 What challenges do you think students may face with learning coding in the classroom (you may choose more than one answer choice)?

- No prior experience (1)
- Lack of interest (2)
- Lack of access to technology (wi-fi, device) (3)
- Perceive it as too difficult (4)
- Other (please specify) (5) ____

Q38 What challenges do you think administration may encounter in offering coding instruction in the classroom (you may choose more than one answer choice)?

- Lack of resources to purchase technology (i.e, student devices, enhanced wi-fi) (1)
- Do not perceive it as being important enough to warrant time and money (2)
- Educators have not been prepared to teach it in the classroom (3)
- Other (please specify) (4) ____

Q39 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding instruction should be required. (1)	☐	☐	☐	☐	☐

Display This Question:
If Please respond to the following statement. = Strongly agree
Or Please respond to the following statement. = Somewhat agree

Q40 What grade levels do you think coding should be instructed (you may choose more than one answer choice)?

- K–2 (1)
- 3–5 (2)
- 6–8 (3)
- 9–12 (4)

Display This Question:
If Please respond to the following statement. = Strongly agree
Or Please respond to the following statement. = Somewhat agree

Q41 How long do you think coding lessons should be per week (in total)?

- 30 minutes (1)
- 45 minutes (2)
- 60 minutes (3)
- over 60 minutes (4)

Q42 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding should be a standalone subject. (1)	○	○	○	○	○

Q43 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Coding should be integrated with other subjects like math, science, and language. (1)	○	○	○	○	○

Q44 Is coding instruction required in your school?

- Yes (1)
- No (2)

Display This Question: If Is coding instruction required in your school? = Yes

Q45 If coding is instructed in your school, why is that the case?

- School-based program (1)
- State requirement (2)
- Other (3) ____

Display This Question: If Is coding instruction required in your school? = Yes

Q46 Identify how coding is taught in your school.

- It is its own separate class (1)
- It is embedded in the instruction of other classes (2)
- It is an extracurricular activity (i.e., robotics) (3)
- Other (4) ____

Q47 How should students learn coding in school (you may choose more than one answer choice)?

- Teacher driven followed by individual student work (1)
- Some teacher guidance followed by collaborative student work (2)
- Teacher is not involved; only student collaborative work (3)
- Other (please specify) (4) ____

Q48 How effectively can you teach block-based coding programs like Scratch?

- Highly capable (1)
- Capable (2)
- Somewhat capable (3)
- Not capable (4)

Q49 How effectively can you teach syntax-based coding programs such as Python?

- Highly capable (1)
- Capable (2)
- Somewhat capable (3)
- Not capable (4)

Q50 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
I can create coding activities at the appropriate level for my students. (1)	o	o	o	o	o

Q51 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
I can integrate coding into my content curriculum. (1)	○	○	○	○	○

Q52 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
A structural change in curriculum is needed to incorporate this new literacy. (1)	○	○	○	○	○

Q53 Please rank the order of your implemented teaching methods (from most used to least).
______ Direct Instruction Only (Lecture) (1)
______ Student cooperative/collaborative learning (2)
______ Direct instruction followed by student application (i.e., student research, formative assessments) (3)
______ Inquiry-based learning (4)
______ Flipped classroom (5)
______ Kinesthetic learning (6)

Q54 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Direct instruction is an effective means to student learning. (1)	○	○	○	○	○

Q55 Please respond to the following statement.

	Strongly Agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Student collaboration and communication are important components of learning. (1)	○	○	○	○	○

Q56 Please respond to the following question.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Learning means remembering what the teacher has taught. (1)	○	○	○	○	○

Q57 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Teaching is presenting or explaining the subject matter. (1)	○	○	○	○	○

Q58 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Technology should be a focal tool in the classroom. (1)	○	○	○	○	○

Q59 Please respond to the following question.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
The focus of teaching is to help students construct knowledge from their learning experience instead of imparting knowledge. (1)	○	○	○	○	○

Q60 Please respond to the following statement.

	Strongly agree (1)	Somewhat agree (2)	Neither agree nor disagree (3)	Somewhat disagree (4)	Strongly disagree (5)
Students should have greater control over their learning process. (1)	○	○	○	○	○

Q61 Constructionism is a theoretical framework developed by renowned educator and mathematician Seymour Papert. He emphasized the use of technology in the classroom as a constructional medium—students collaborate to create products and then communicate the results. Based on this statement, please respond to the following statement.

	Easily possible (1)	Possible but difficult to do (2)	Not possible (3)
This framework is possible to apply in the classroom. (1)	○	○	○

Q62 What is your gender identification?

- Male (1)
- Female (2)
- Nonbinary (3)
- Other (4)
- Decline to state (5)

Q63 Please choose your age range.

- 20–24 (1)
- 25–29 (2)
- 30–34 (3)
- 35–39 (4)
- 40–44 (5)
- 45–49 (6)
- 50–54 (7)
- 55–59 (8)
- 60–64 (9)
- 65–69 (10)
- 70–74 (11)
- 75–79 (12)
- 80–84 (13)
- 85 and above (14)

Q64 What is your highest level of education?

- Associate in Arts degree (1)
- Bachelor's degree (2)
- Master's degree (3)
- Doctoral degree (EdD or PhD) (4)
- Other (please identify) (5) ____

Q65 What is your role at school (you may choose more than one answer choice)?

- o Teacher (1)
- o Administrator (i.e., principal, vice-principal) (2)
- o STEM director/technology coordinator (3)
- o Multiple roles (please identify) (4) ____

Q66 What grade level do you instruct (you may choose more than one answer choice)?

- o K–2 (1)
- o 3–5 (2)
- o 6–8 (3)
- o 9–12 (4)
- o I do not teach (i.e., I am an administrator only) (5)

Q67 What subject/s do you instruct?

- o Mathematics (1)
- o Science (2)
- o History/Social Studies (3)
- o English (4)
- o Reading (5)
- o Music (6)
- o Art (7)
- o Foreign Language (8)
- o Religion (10)
- o Other (please indicate) (13) ____
- o I do not teach in the classroom (14)

Q68 What is your religious affiliation?

- o Christian (1)
- o Jewish (2)
- o Muslim (3)
- o Other (please indicate) (4) ____

Q69 What is your racial background?

- o American Indian or Alaska Native not Hispanic (1)
- o Asian not Hispanic (2)
- o Pacific Islander Not Hispanic (3)
- o Filipino not Hispanic (4)
- o Hispanic or Latino of any race (5)
- o African American not Hispanic (6)
- o White not Hispanic (7)
- o Two or more races not Hispanic (8)
- o Other (please indicate) (9) ____

Q70 How long have you served in the education profession?

- o Less than 1 year (1)
- o 1–2 years (2)
- o 2–4 years (3)
- o 4–6 years (4)
- o More than 6 years (5)
- o Don't know/not sure (6)

Q71 Do you hold a California teaching credential?

- o Yes (1)
- o No (2)

Display This Question: If Do you hold a California teaching credential? = Yes

Q72 What type of credential do you hold?

- o Multiple subject (1)
- o Single subject (please identify) (2) ____

Q73 If you would like to enter into a raffle to win one of five $25 Amazon gift cards, please enter your email address below.

____

APPENDIX L

Table of Specifications for Survey

Theme	Number of Questions	Specific Questions
Informed Consent	1	1
Instructions to Survey	1	2
Understanding of Coding	18	3–20
Benefits of Coding	15	21–35
Implementation of Coding	17	36–52
Epistemology	9	53–61
Demographics	11	62–72
Total	72	

APPENDIX M

Survey Assessment Rubric

Thank you so much for helping me with my survey development!

1. The purpose of this survey is to …
2. The target population will be …
3. This survey is intended to provide data for the following Research Questions: [list here]
4. Please take the survey in its entirety. Note the item numbers of any questions that are ambiguous and/or confusing. On completion of the survey, complete this survey assessment rubric. Narrative comments of strengths and weaknesses, along with suggestions for improvement are of great benefit.

Element	Standard	Comments
Survey Instructions	• Purpose of survey is clear. • Instructions to participants are clear and easy to follow.	
Survey Organization/Layout	• The layout of the survey makes it easy to read and respond to all items	
Survey Language	• Clear, unambiguous, specific language is utilized throughout	
Survey Content Appropriate for Intended Audience	• Survey instructions, items, and demographic questions are appropriate for the target population. • Survey items have a high likelihood of providing data to address research questions.	
Survey Face Validity	• All items on the survey appear to be relevant to the purpose	
Survey Demographics	• Relevant demographic information is asked for • Demographic information is placed appropriately in the survey	
Survey Number of Items	• Contains adequate number and types of Likert scale items and open-ended items to address research questions.	
Survey Items that Need Revising	• List question number(s) that are ambiguous or confusing	

Strengths of Survey:

Weaknesses of Survey & Suggestions for Improvement:

APPENDIX N

IBM SPSS Privacy Policy

IBM (2022) SPSS Statistics & IBM SPSS Amos & IBM SPSS Data Access Pack & IBM SPSS Text Analytics for Surveys Considerations for GDPR Readiness

Last Updated: 2021-03-22

Information about features of SPSS® Statistics that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness.

For PID(s): 5725-A54, 5725-A60, 5725-A56, 5725-A62

Notice:

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of SPSS Statistics that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party applications and systems.

Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.

The products, services, and other capabilities described herein are not suitable for all client situations and may have restricted availability. IBM® does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.

Table of contents

GDPR

General Data Protection Regulation (GDPR) has been adopted by the European Union and will apply from May 25, 2018.

Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

Read more about GDPR:

- EU GDPR Information Portal
- ibm.com/GDPR web site

Offering Configuration

The following sections provide considerations for configuring SPSS Statistics to help your organization with GDPR readiness.

The following configuration steps are recommended for compliance with GDPR regulations if personal data is to be stored in SPSS Statistics:

- Activate the encryption of data in motion (data that is transferred over a data network between SPSS Statistics client applications and SPSS Statistics server; data between SPSS Statistics server and external execution servers—for example, IBM SPSS Collaboration and Deployment Services).
- Do not share passwords for the operation of SPSS Statistics between multiple persons.

Data Life Cycle

GDPR requires that personal data is:

- Processed lawfully, fairly and in a transparent manner in relation to individuals.
- Collected for specified, explicit and legitimate purposes.
- Adequate, relevant and limited to what is necessary.
- Accurate and, where necessary, kept up to date. Every reasonable step must be taken to ensure that inaccurate personal data are erased or rectified without delay.
- Kept in a form which permits identification of the data subject for no longer than necessary.

What is the end-to-end process through which personal data go through when using SPSS Statistics?

What types of data flow through SPSS Statistics?

SPSS Statistics is a general purpose data analytics software offering that provides clients with the ability to manipulate and analyze data for your business needs. There are two deployment approaches, and most customers use the client software on its own. It operates on your own desktop computer without connections to other parties.

Aside from the client software, we also offer a client-server architecture that allows customers to connect to and operate data on a server environment.

SPSS Statistics software does not store any personal data. You have total control of how your data is handled, used, or stored. The data may or may not include personal user data. SPSS Statistics acts as a general tool—only processing the data based on your business requirements. Your data are transparent to IBM and we have no intention to use it. Only for debugging and auditing purposes, the personal information that SPSS Statistics collects includes the user ID used to log in to SPSS Statistics Server, the IP address for the connection, and the database connection information.

IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle data the same way as SPSS Statistics. These tools don't store any personal data. You have complete control over how your data is handled, used, or stored.

Where in the process?

- Authentication: For SPSS Statistics client deployment, IBM doesn't provide authentication. Who can access the software depends on the native operating system control. For SPSS Statistics client-server deployment, all operations to SPSS Statistics server must be authenticated. When a user connects to SPSS Statistics server via SPSS Statistics client, they must log on to the SPSS Statistics server with an authenticated user ID on the operating system where the SPSS Statistics server is running. SPSS Statistics prompts the user to enter their password once the connection is closed (for example, when SPSS Statistics client is restarted), and SPSS Statistics doesn't store the password.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't provide authentication. The native operating system controls who can access the software.
- Logging: For SPSS Statistics client deployment, logging includes a journal file and a trace log. The journal file records instrumentation of user operations including SPSS Statistics launch time, exit time, command executed, and timestamp. The trace log collects application usage data as logs for diagnostic purposes. Operation system authorized users can enable/disable the logs or change the log level based on his/her needs. For SPSS Statistics client-server deployment, logging includes a journal file and a server log. The journal file records instrumentation of user operations including SPSS Statistics launch time, exit time, command executed, and timestamp. The server log is meant to provide benchmarking data that IBM can provide for our customers and to help SPSS Statistics Server administrators diagnose performance problems. Operation system authorized users can enable/disable the logs or change the log level based on his/her needs.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle logging in a similar way as SPSS Statistics client.
- Configuration: Statistics configuration is stored in a configuration file that is restricted by file-level authentication and encryption.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys treat configuration the same way as SPSS Statistics.

For what purpose?

SPSS Statistics software does not store any personal data. You have total control of how your data is handled, used, and stored. The data may or may not include personal user data. IBM acts as a general tool, only processing the data based on your business requirements. Your data are transparent to IBM and we have no intentions of using it.

IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle data the same way as SPSS Statistics. These tools don't store any personal data. You have complete control over how your data is handled, used, or stored.

Personal data used for online contact with IBM

SPSS clients can submit online comments/feedback/requests to contact IBM about SPSS subjects in a variety of ways, primarily:

- Public comments area on pages in the SPSS community on IBM developerWorks®
- Public comments area on pages of SPSS product documentation in IBM Knowledge Center
- Public comments in the SPSS space of dWAnswers
- Feedback forms in the SPSS community

Typically, only the client name and e-mail address are used, to enable personal replies for the subject of the contact, and the use of personal data conforms to the IBM Online Privacy Statement.

When assessing your use of SPSS Statistics and the requirements of GDPR, you should consider the types of personal data which in your circumstances are passing through SPSS Statistics.

Aside from logging features, SPSS Statistics software doesn't collect any data. You have full control of how your data is handled, used, and stored. Your data may or may not include personal user data. IBM acts as a general tool, only processing your data based on your business requirements. Customer data are transparent to IBM and we have no intentions of using it.

Data Storage

The following Data Storage mechanisms are used by SPSS Statistics, which users may wish to consider when assessing their GDPR readiness.

- Storage of account data: SPSS Statistics software does not store any account data. You have total control of how your data is handled, used, and stored. The data may or may not include personal user data. IBM acts as a general tool, only processing the data based on your business requirements. Your data are transparent to IBM and we have no intentions of using it. IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle data the same way as SPSS Statistics. These tools don't store any personal data. You have complete control over how your data is handled, used, or stored.
- Storage of client data: SPSS Statistics software does not store any client data. You have total control of how your data is handled, used, and stored. The data may or may not include personal user data. IBM acts as a general tool, only processing the data based on your business requirements. Your data are transparent to IBM and we have no intentions of using it. IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle data the same way as SPSS Statistics. These tools don't store any personal data. You have complete control over how your data is handled, used, or stored.
- Storage in backups: SPSS Statistics software does not store any backup data. You have total control of how your data is handled, used, and stored. The data may or may not include personal user data. IBM acts as a general tool, only processing the data based on your business requirements. Your data are transparent to IBM and we have no intentions of using it. IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle data the same way as SPSS Statistics. These tools don't store any personal data. You have complete control over how your data is handled, used, or stored.
- Storage in archives: SPSS Statistics software does not store any archived data. You have total control of how your data is handled, used, and stored. The data may or may not include personal user data. IBM acts as a general tool, only processing the data based on your business requirements. Your data are transparent to IBM and we have no intentions of using it.

IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle data the same way as SPSS Statistics. These tools don't store any personal data. You have complete control over how your data is handled, used, or stored.

Data Access

Who can access data in SPSS Statistics?

- SPSS Statistics doesn't control access to data. Access control to personal data depends on operating system permissions to files generated by SPSS Statistics (for example,.sav files, log files, and preference files).
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't control access to data. Access control to personal data depends on operating system permission to files generated by those tools (for example, amw files, log files, and preference files).
- Roles and access rights: SPSS Statistics doesn't manage any roles or access rights by itself. The role definitions and access control rights can be handled by the operating system.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't manage any roles or access rights. Role definition and access control can be handled by the operating system.
- Separation of duties: SPSS Statistics does not manage any duties separation by itself. The duty definitions and access control rights can be handled by the operating system.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't have any privileged administrators. Role definition and access control rights can be handled by the operating system.
- Privileged Administrators: SPSS Statistics doesn't have any privileged administrators, and role definitions and access control rights can be handled by the operating system.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys manage any roles or access rights. Role definition and access control can be handled by the operating system.
- Administrators: SPSS Statistics doesn't have any administrators, and role definitions and access control rights can be handled by the operating system.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't have any administrators. Role definition and access control can be handled by the operating system.
- Activity logs: SPSS Statistics doesn't control access to log data. Access control can be handled by the operating system. An authorized operating system user can enable/disable the logs or change the log level based on his/her needs.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't control access to log data. Access control can be handled by the operating system. Authorized operating system users can enable/disable the logs or change the log level based on his/her needs.

Data Processing

- Encryption in motion: SSL is supported by SPSS Statistics client-server deployments. This does not apply to SPSS Statistics client-only deployments.
- Encryption at rest: You can choose to save SPSS Statistics data files (.sav) with a password. Gskit is used to support this functionality.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't provide encryption. Encryption can be handled by other tools or the operating system.

- Encryption key ownership: You own the encryption key. SPSS Statistics doesn't own, control, or store the encryption key.

Data Deletion

Right to Erasure

Article 17 of the GDPR states that data subjects have the right to have their personal data removed from the systems of controllers and processors—without undue delay—under a set of circumstances.

Data Deletion Characteristics

- Client data deletion: You have total control of how your data is handled, used, and stored. The data may or may not include personal user data. Some features of SPSS Statistics can be used to clear data or overwrite data. But this is controlled by you.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle data the same way as Statistics. These tools don't store any personal data. You have complete control over how your data is handled, used, or stored.
- Account data deletion: You have total control of how your data is handled, used, and stored. The data may or may not include personal user data. Some features of SPSS Statistics can be used to clear data or overwrite data. But this is controlled by you.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys handle data the same way as Statistics. These tools don't store any personal data. You have complete control over how your data is handled, used, or stored.

Data Monitoring

You should regularly test, assess, and evaluate the effectiveness of your technical and organizational measures to comply with GDPR. These measures should include ongoing privacy assessments, threat modeling, centralized security logging, and monitoring, among others.

Responding to Data Subject Rights

Will I be able to address Data Subject requests from my customers?

- Right to Access: SPSS Statistics doesn't control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
- Right to Modify: SPSS Statistics doesn't control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
- Right to Restrict Processing: SPSS Statistics doesn't control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don't control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.

- Right to Object: SPSS Statistics doesn’t control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don’t control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
- Right to Be Forgotten: SPSS Statistics doesn’t control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don’t control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
- Right to Data Portability: SPSS Statistics doesn’t control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.
  IBM SPSS Amos, IBM SPSS Data Access Pack, and IBM SPSS Text Analytics for Surveys don’t control access to data. Access control to personal data depends on operating system permissions that you configure. You have total control.

REFERENCES

Adams, A., & Mowers, H. (2013, October 30). Should coding be the "new foreign language" requirement? *Edutopia*. https://www.edutopia.org/blog/coding-new-foreign-language-requirement-helen-mowers

Alanazi, A. (2016). A critical review of constructivist theory and the emergence of Constructionism. *American Research Journal of Humanities and Social Sciences, 2*, 1–8. https://doi.org/10.21694/2378-7031.16018

Alexiou-Ray, J., Raulston C., Fenton, D., & Johnston, S. (2018). Coding in the K–12 classroom. *The K–12 educational technology handbook*. Ed Tech Books. https://edtechbooks.org/k12handbook/coding_in_k-12

Armoni, M. (2016). Computer science, computational thinking, programming, coding: The anomalies of transitivity in K—12 computer science education. *ACM Inroads, 7*(4), 24–27. https://doi.org/10.1145/3011071

Asunda, P. A. (2012). Standards for technological literacy and STEM education delivery through career and technical education programs. *Journal of Technology Education 23*(2), 44–60. (ERIC Document Reproduction Service No. EJ976765). https://eric.ed.gov/?id=EJ976765

Avendano, L., Renteria, J., Kwon, S., & Hamdan, K. (2018). Bringing equity to underserved communities through STEM education: Implications for leadership development. *Journal of Education Administration and History, 51*(1), 66–82. https://doi.org/10.1080/00220620.2018.1532397

Badilla-Saxe, E. (2010, August 16–20). *Constructionism, complex thinking and emergent learning: Preschool children designing and programming* [Paper presentation]. Constructionism 2010, Paris, France.

Baker-Doyle, K. J. (2018). I, pseudocoder: Reflections of a literacy teacher-educator on teaching coding as critical literacy. *Contemporary Issues in Technology and Teacher Education, 18*(2), 255–270. https://www.learntechlib.org/primary/p/180585/

Banzato, M., & Tosato, P. (2017). An exploratory study of the impact of self-efficacy and learning engagement in coding learning activities in Italian middle school. *International Journal on E-Learning, 16*(4), 349–369. https://www.learntechlib.org/p/152245

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K—12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 111–122. https://doi.org/10.1145/1929887.1929905

Beisser, S., & Gillespie, C. (2003). Kindergarteners can do it—so can you: A case study of a constructionist technology-rich first year seminar for undergraduate college students. *Information Technology in Childhood Education Annual*, *2003*(1), 243–260. https://www.learntechlib.org/primary/p/17766/

Bell, T. (2016, August 08). *What's all the fuss about coding?* [Paper presentation]. ACER Research Conference 2016, Brisbane. https://research.acer.edu.au/research_conference/RC2016/8august/12

Bers, M. U. (2018). Coding and computational thinking in early childhood: The impact of Scratch Jr in Europe. *European Journal of STEM Education, 3*(3), 1–13. https://doi.org/10.20897/ejsteme/3868

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and Tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education, 72*, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Bers, M. U., & Sullivan, A. (2019). Computer science education in early childhood: The case of ScratchJr. *Journal of Information technology Education: Innovations in Practice 18,* 113–138. https://doi.org/10.28945/4437

Beynon, M., & Roe, C. (Eds.). (2004). Computer support for constructionism in context. *IEEE International Conference on Advanced Learning Technologies.* IEEE Xplore. https://doi.org/10.1109/ICALT.2004.1357406

Bilkstein, P. (2018). *Pre-college computer science education: A survey of the field.* Google LLC. https://services.google.com/fh/files/misc/pre-college-computer-science-education-report.pdf

Blustein, D. L., Barnett, M., Mark, S., Depot, M., Lovering, M., & Lee, Y., Hu, Q., Kim, J., Backus, F., Dillon-Lieberman, K., & DeBay, D. (2013). Examining urban students' constructions of a STEM/career development intervention over time. *Journal of Career Development, 40*(1), 40–67. https://doi.org/10.1177/0894845312441680

Burke, Q., O'Byrne, I., & Kafai, Y. (2016). Computational participation: Understanding coding as an extension of literacy instruction. *Journal of Adolescent and Adult Literacy, 59*(4), 371–375. https://doi.org/10.1002/jaal.496

Burnett, C., & Merchant, G. (2015). The challenge of the 21st century literacies. *Journal of Adolescent and Adult Literacy, 59*(3), 271–274. https://doi.org/10.1002/jaal.482

Cakir, N. K., & Guven, G. (2019). Arduino-assisted robotic and coding applications in science teaching: Pulsimeter activity in compliance with the 5E learning model *Science Activities, 56*(2), 42–51. https://doi.org/10.1080/00368121.2019.1675574

Calabrese-Barton, A., & Tan, E. (2018). A longitudinal study of equity-oriented STEM-rich making among youth from historically marginalized communities. *American Educational Research Journal, 55*(4), 761–800. https://doi.org/10.3102/0002831218758668

Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom, 15*(4), 9–14. http://www.aamt.edu.au/Webshop/Entire-catalogue/Australian-Primary-Mathematics-Classroom

Calder, N. (2018). Using Scratch to facilitate mathematical thinking. *Waikato Journal of Education, 23*(2), 43–58. https://doi.org/10.15663/wje.v23i2.654

Casserly, M. (2012, May 11). 10 jobs that didn't exist 10 years ago. *Forbes.* https://www.forbes.com/sites/meghancasserly/2012/05/11/10-jobs-that-didnt-exist-10-years-ago/

Cengel, M., Alkan, A., & Cayir, E. (2018, July 18–20). *In the determination of self-efficacy situations of information technology teachers' coding for middle school students, robotic coding and 3dimensional design examples of Sakarya provinc*e [Paper presentation]. International Conference on New Horizons in Education (INTE), Paris, France.

Chan, K. (2004). Preservice teachers' epistemological beliefs and conceptions about teaching and learning: Cultural implications for research in teacher education. *Australian Journal of Teacher Education, 29*(1), 1–13. https://doi.org/10.14221/ajte.2004v29n1.1

Chan, K., & Elliot, R. G. (2004). Relational analysis of personal epistemology and conceptions about teaching and learning. *Teaching and Teacher Education, 20*(8), 817–831. https://doi.org/10.1016/j.tate.2004.09.002

Chang, Y. H., & Peterson, L. (2018). Pre-service teachers' perceptions of computational thinking. *Journal of Technology and Teacher Education, 26*(3), 353–374. https://www.learntechlib.org/primary/p/181433/

Code.org Advocacy Coalition. (2019). *2019 state of computer science education.* https://advocacy.code.org/2019_state_of_cs.pdf

Code.org. (n.d.-a). Code.org cs fundamentals—glossary.https://code.org/curriculum/docs/k-5/glossary

Code.org. (n.d.-b). *Make computer science in K—12 count!* https://code.org/files/convince_your_school_or_state.pdf

Common Core State Standards Initiative. (2020). *Standards in your state.* http://www.corestandards.org/standards-in-your-state/

Corradini, I., Lodi, M., & Nardelli, E. (2018, October). *An investigation of Italian primary school teachers' view on coding and programming* [Paper presentation]. Fundamentals of Computer Science and Software Engineering—11th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, St. Petersburg, Russia. https://doi.org/10.1007/978-3-030-02750-6_18

Culp, K. M., Honey, M., & Mandinach, E. (2003). *A retrospective on twenty years on education technology policy.* American Institutes for Research. https://www2.ed.gov/rschstat/eval/tech/20years.pdf

Cutumisu, M., & Guo, Q. (2019). Using topic modeling to extract pre-service teachers' understandings of computational thinking from their coding reflections. *IEEE Transactions on Education*, *62*(4), 325–332. https://doi.org/10.1109/TE.2019.2925253

Darling-Hammond, L., Burns, D., Campbell, C., Goodwin, A. L., Hammerness, K., Low, E. L., McIntyre, A., Sato, M., & Zeichner, K. (2017). *Empowered educators: How high-performing systems shape teaching quality around the world*. Jossey-Bass Publishers.

DePryck, K. (2016). From computational thinking to coding and back. In F. J. García-Peñalvo (Ed.), Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'16) (Salamanca, Spain, November 2–4, 2016) (pp. 27–29). New York, NY, USA: ACM. https://doi.org/10.1145/3012430.3012492

diSessa, A. A. (2018). Computational literacy and "the big picture" concerning computers in mathematics education. *Mathematical Thinking and Learning, 20*(1), 3–31. https://doi.org/10.1080/10986065.2018.1403544

Duncan, C., Bell, T., & Tanimoto, S. (2014). Should your 8-year-old learn coding? *WiPSCE '14: Proceedings of the 9th Workshop in Primary and Secondary Computing Education*. https://doi.org/10.1145/2670757.2670774

Estapa, A., Hutchison, A., & Nadolny, L. (2018). Recommendations to support computational thinking in the elementary classroom. *Technology and Engineering Teacher, 77*(4), 25–29. https://www.iteea.org/Publications/Journals/TET/TETDecJan2018.aspx

Falloon, G. (2016). An analysis of young students' thinking when completing basic coding tasks using Scratch Jnr. on the iPad. *Journal of Computer Assisted Learning, 32*(6), 576–593. https://doi.org/10.1111/jcal.12155

Farooq, M. S., Abid, A, Khan, S. A., Naem, M. A., Farooq, A., Abid, K., & Shafiq, M. (2012). A qualitative framework for introducing programming language at high school. *Journal of Quality and Technology Management, 8*(2), 135–151.

Fields, D., Vasudevan, V., & Kafai, Y. B. (2015). The programmers' collective: Fostering participatory culture by making music videos in a high school Scratch coding workshop. *Interactive Learning Environments*, *23*(5), 613–633. https://doi.org/10.1080/10494820.2015.1065892

Fincher, S., & Utting, I. (2010). Machines for thinking. *ACM Transactions on Computing Education, 10*(4), 1–7. https://doi.org/10.1145/1868358.1868360

Frey, C. B., & Osborne, M. A. (2013). *The future of employment: How susceptible are jobs to computerization.* Oxford University Engineering Department Machines and Employment Workshop. https://www.oxfordmartin.ox.ac.uk/downloads/academic/The_Future_of_Employment.pdf

García-Peñalvo, F. J., Reimann, D., & Maday, C. (2018). Introducing coding and computational thinking in the schools: The TACCLE 3—coding project experience. In M. S. Khine (Ed.), *Computational thinking in the STEM disciplines: Foundations and research highlights* (pp. 213–226). Springer. http://doi.org/10.1007/978-3-319-93566-9_11

García-Peñalvo, F. J., Ress, A. M., Hughes, J., Jormanainen, I., Toivonen, T., & Vermeersch, J. (2016, November). *A survey of resources for introducing coding into schools* [Paper presentation]. TEEM '16: Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain.

Gee, J. P. (2013). *Good video games and good learning: Collected essays on video games, learning, and literacy* (2nd ed.). Peter Lang.

Google. (2015). *Images of computer science: Perceptions among students, parents and educators in the U.S.* Voced Plus. https://www.voced.edu.au/content/ngv%3A71827

Google Inc., & Gallup Inc. (2016). *Trends in the state of computer science in U.S. K–12 schools.* http://services.google.com/fh/files/misc/trends-in-the-state-of-computer-science-report.pdf

Gove, M. (2014). Michael Gove speaks about computing and educational technology. *Speech to the BETT Conference, London.* [Online]. Gov.UK. https://www.gov.uk/government/speeches/michael-gove-speaks-about-computing-and-education-technology

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Gunbatar, M. S., & Karalar, H. (2018). Gender differences in middle school students' attitudes and self-efficacy perceptions towards mBlock programming. *European Journal of Educational Research 7*(4), 925–933. https://doi.org/10.12973/eu-jer.7.4.923

Hagge, J. (2018). Coding to create: A subtext of decisions as early adolescents design digital media. *Technology, Knowledge, and Learning, 23*(2), 247–271. https://doi.org/10.1007/s10758-018-9359-y

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments, 1*(1), 1–32. http://www.iditharel.com/wp-content/uploads/2014/09/Harel-Papert-Paper.pdf

Hargreaves, A., & Fullan, M. (2020). Professional capital after the pandemic: Revisiting and revising classic understandings of teachers' work. *Journal of Professional Capital and Community, 5*(3/4), 327–336. https://doi.org/10.1108/JPCC-06-2020-0039

Harlow, D. B., Dwyer, H., Hansen, H. K., Hill, C., Iveland, A., Leak, A. E., & Franklin, D. M. (2016). Computer programming in elementary and middle school: Connections across content. In M. Urban, & D. Falvo (Eds.) *Improving K–12 STEM education outcomes through technological integration* (pp. 337–361). IGI Global. https://doi.org/10.4018/978-1-4666-9616-7.ch015

Hutchison, A., Nadolny, L., & Estapa, A. (2015). Using coding apps to support literacy instruction and develop coding literacy. *The Reading Teacher, 69*(5), 493–503. https://doi.org/10.1002/trtr.1440

IBM. (2022). *Statistical package for the social sciences* (Version 27). https://www.ibm.com/analytics/spss-statistics-software

Israel, M., Wherfel, Q. M., Pearson, J. N., Shehab, S., & Tapia, T. (2015). Empowering K-12 students with disabilities to learn computational thinking and computer programming. *Teaching Exceptional Children, 48*(1), 45–53. https://doi.org/10.1177/0040059915594790

Jancheski, M. (2017). Improving teaching and learning computer programming in schools through educational software. *Olympiads in Informatics, 11,* 55–75. https://doi.org/10.15388/ioi.2017.05

Johnson-Bailey, J., & Cervero, R. M. (2004). Mentoring in Black and White: The intricacies of cross-cultural mentoring. *Mentoring & Tutoring: Partnership in Learning, 12*(1), 7–21. https://doi.org/10.1080/1361126042000183075

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming.* MIT Press.

Kafai, Y. B., Fields, D. A., & Searle, K. A. (2014). Electronic textiles as disruptive designs: Supporting and challenging maker activities in schools. *Harvard Educational Review, 84*(4), 532–556. https://doi.org/10.17763/haer.84.4.46m7372370214783

Keane, T., Chalmers, C., Boden, M., & Williams, M. (2019). Humanoid robots: Learning a programming language to learn a traditional language. *Technology, Pedagogy, & Education*, *28*(5), 533–546. https://doi.org/10.1080/1475939X.2019.1670248

Kim, G. W. (2019). A learning model for software coding education. *Journal of Problem-Based Learning, 6*(2), 67–75. https://doi.org/10.24313/jpbl.2019.00164

King, A. (2015). Reflecting on classroom practice: Spatial reasoning and simple coding. *Australian Math Teacher, 71*(4), 21–27. http://www.aamt.edu.au/Webshop/Entire-catalogue/Australian-Mathematics-Teacher

Kong, R., & Wong, G. K. W. (2017, December 12–14). *Teachers' perception of professional development in coding education* [Paper presentation]. IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Hong Kong, China.

Kuchey, D., & Flick, M. (2017). Contest corner: Narrowing the achievement gap with C-STEM curriculum. *Ohio Journal of School Mathematics, 75*(1), 49–53. https://library.osu.edu/ojs/index.php/OJSM/article/view/5614

K12 Computer Science. (n.d.). *Equity in computer science education*. https://k12cs.org/equity-in-computer-science-education/

Lambert, D. (2019). California moves to get more K–12 students into computer science classes. *EdSource*. https://edsource.org/2019/california-moves-to-get-more-k-12-students-into-computer-science-classes/612158

Lane, M., Hui, W., Murcia, K., Wongthongtham, P. (2019, February 8–10). *Diversity Implications of Online Learning of Coding* [Paper presentation]. International Association for Development of the Information Society (IADIS) International Conference on Educational Technologies, Hong Kong, China, and Wongthongtham (2019).

Lavadenz, M. & Kaminski, L. (2020, Summer). *Contexts and current topics in public education.* School of Education. Loyola Marymount University, Los Angeles.

Leavy, P. (2017). *Research design: Quantitative, qualitative, mixed methods, arts-based, and community-based participatory research approaches.* The Guilford Press.

Litts, B. K., Kafai, Y. B., Lui, D. A., Walker, J. T., & Widman, S. A. (2017). Stitching codable circuits: High school students' learning about circuitry and coding with electronic textiles. *Journal of Science Education and Technology, 26*(5), 494–507. https://doi.org/10.1007/s10956-017-9694-0

Livingstone, S. (2012). Critical reflections on the benefits of ICT in education. *Oxford Review of Education, 38*(1), 9–24. https://doi.org/10.1080/03054985.2011.577938

Logo Foundation. (n.d.). *Logo history.* Retrieved August 20, 2022, from https://el.media.mit.edu/logo-foundation/what_is_logo/history.html

Lui, D., Kafai, Y., Litts, B., Walker, J., & Widman, S. (2020). Pair physical computing: High school students' practices and perceptions of collaborative coding and crafting with electronic textiles. *Computer Science Education, 30*(1), 72–101. https://doi.org/10.1080/08993408.2019.1682378

Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education, 19*(4), 790–824. https://citejournal.org/volume-19/issue-4-19/general/preparing-elementary-school-teachers-to-teach-computing-coding-and-computational-thinking

Mason, S. L., & Rich, P. J. (2020). Development and analysis of the Elementary Student Coding Attitudes Survey. *Computers and Education, 153*, Article 103898 https://doi.org/10.1016/j.compedu.2020.103898

Master, A., & Meltzoff, A. N. (2017). Building bridges between psychological science and education: Cultural stereotypes, STEM, and equity. *Prospects: Quarterly Review of Comparative Education, 46*(2), 215–234. https://doi.org/10.1007/s11125-017-9391-z

McCoy, D. L., Luedke, C. L., & Winkle-Wagner, R. (2017). Encouraged or weeded out: Perspectives of students of color in the STEM disciplines on faculty interactions. *Journal of College Student Development, 58*(5), 657–673. https://doi.org/10.1353/csd.2017.0052

McCoy-Parker, K. S., Paull, L. N., & Montgomery, S. E. (2017). Challenging elementary learners with programmable robots during free play and direct instruction. *Journal of STEM Arts, Crafts, and Constructions, 2*(2), 100–129.

McCullough, M. (2020, Fall). *Organizational theory and change.* School of Education. Loyola Marymount University, Los Angeles.

Mills, G. E., & Gay, L. R. (2019). *Educational research*. Pearson.

National Center for Education Statistics. (2018). *Degrees in computer and information sciences conferred by postsecondary institutions, by level of degree and sex of student: 1970–71 through 2016–17*. https://nces.ed.gov/ipeds/Search?query=computer%20science%20bachelor%27s%20degrees&query2=computer%20science%20bachelor%27s%20degrees&resultType=all&page=1&sortBy=relevance&overlayDigestTableId=200727

National Defense Education Act of 1958, Pub. L. No. 85-864, 72 Stat. 1580 (1958). https://history.house.gov/HouseRecord/Detail/15032436195

NortonLifelock. (2022). *Norton 360 Deluxe* (2022). https://us.norton.com/products/norton-360-deluxe

Noss, R., & Clayson, J. (2015). Reconstructing constructionism. *Constructivist Foundations, 10*(3), 285–288. https://constructivist.info/10/3/285.noss

Owens, R. G., & Valesky, T. C. (2015). *Organizational behavior in education: Leadership and school reform*. Pearson.

Papert, S. (1999). Logo philosophy and implementation. *Logo Computer Systems Inc*. http://www.microworlds.com/company/philosophy.pdf

Papert, S., & Harel, I. (1991). *Constructionism*. Ablex Publishing Company.

Perry, C. (2015). *Coding in schools*. Northern Ireland Assembly. http://www.niassembly.gov.uk/globalassets/documents/raise/publications/2015/education/3715.pdf

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education, 128*, 365–376. https://doi.org/10.1016/j.compedu.2018.10.005

Price, C. B., & Price-Mohr, R.M. (2018). An evaluation of primary school children coding using a text-based language (Java). *Computers in the Schools, 35*(4), 284–301. https://doi.org/10.1080/07380569.2018.1531613

Project Tomorrow. (2018). *Coding in K–8 classrooms: Empowering a new generation of creators.* (ERIC Document Reproduction Service No. D600542). https://files.eric.ed.gov/fulltext/ED600542.pdf

PR Newswire. (2015, October 5). *Horizon media study reveals Americans prioritize STEM subjects over the arts; Science is "cool," coding is new literacy*. https://www.prnewswire.com/news-releases/horizon-media-study-reveals-americans-prioritize-stem-subjects-over-the-arts-science-is-cool-coding-is-new-literacy-300154137.html

Qualtrics. (2022). *Qualtrics XM* (June 2021-March 2022). https://www.qualtrics.com.

Resnick, M. (2012, November). *Let's teach kids to code* [Video]. TED Conferences. https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=en

Resnick, M., & Siegel, D. (2015). A different approach to coding. *International Journal of People-Oriented Programming*, *4*(1), 1–4. https://doi.org/10.4018/IJPOP.2015010101

Rich, P. J., Browning, S., Perkins, M., Shoop, T., Belikov, O., & Yoshikawa, E. (2017). Coding in K–8: International trends in computing education with primary-aged children. *Teacher Practices in Computational Thinking (Paper Submission)*, 1–32. https://doi.org/10.13140/RG.2.2.29782.14409/1

Rich, P. J., Larsen, R. A., & Mason, S. L. (2020). Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education, 53*(3), 296–316. https://doi.org/10.1080/15391523.2020.1771232

Rich, P. J., Leatham, K. R., & Wright, G. A. (2013). Convergent cognition. *Instructional Science, 41*(2). 431–453. https://doi.org/10.1007/s11251-012-9240-7

Rob, M., & Rob, F. (2018). Dilemma between constructivism and constructionism: Leading to the development of a teaching-learning framework for student engagement and learning. *Journal of International Education in Business, 11*(2), 273–290. https://doi.org/10.1108/JIEB-01-2018-0002

Rossano, V., Roselli, T. & Quercia, G. (2018, October 21–23). *Coding and computational thinking with Arduino* [Paper presentation]. International Association for Development of the Information Society (IADIS) International Conference on Cognition and Exploratory Learning in the Digital Age (CELDA), Budapest, Hungary.

Roumell, E. A., & Salajan, F. D. (2016). The evolution of U.S. e-Learning policy: A content analysis of the National Education Technology Plan. *Educational Policy, 30*(2), 365–397. https://doi.org/10.1177/0895904814550070

Savage, T., Sánchez, I. A., O'Donnell, F., & Tangney, B. (2003, July 9–11). *Using robotic technology as a constructionist mindtool in knowledge construction* [Conference session]. 2003 IEEE International Conference on Advanced Learning Technologies, Athens, Greece. https://doi.org/10.1109/ICALT.2003.1215105

Schommer-Aikins, M. (2004). Explaining the epistemological belief system: Introducing the embedded systemic model and coordinated research approach. *Educational Psychologist, 39*(1), 19–29. https://doi.org/10.1207/s15326985ep3901_3

Sentance, S., & Csizmadia, A. (2016). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies, 22*, 469–495. https://doi.org/10.1007/s10639-016-9482-0

Seow, P., Looi, C. -K., Wadhwa, B., Wu, L., & Liu, L. (2017). Computational thinking and Coding initiatives in Singapore. In S. -C. Kong, J., & R. K. -Y. Li (Eds.), *Conference on computational thinking education* (pp. 164–167). The Education University of Hong Kong.

Serafini, G. (2011, October 26-29). *Teaching programming at primary schools: Visions, experiences, and long-term research prospects* [Paper presentation]. In I. Kalaš, & R. T. Mittermeir (Eds) *Informatics in schools. Contributing to 21st century education* (pp. 143–154). Springer. https://doi.org/10.1007/978-3-642-24722-4_13

Smith, M. (2016, January 30). *Computer science for all.* https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Soykan, F., & Kanbul, S. (2018). Analyzing K12 students' self-efficacy regarding coding education. *TEM Journal, 7*(1), 182–287. https://doi.org/10.18421/TEM71-22

Sparks, D. M., & Pole, K. (2019). Do we teach subjects or students? Analyzing science and mathematics teacher conversations about issues of equity in the classroom. *School Science and Mathematics, 119*(7), 405–416. https://doi.org/10.1111/ssm.12361

Stager, G. (2001, June 25–27). *Constructionism as a high-tech intervention strategy for at-risk learners* [Conference session]. National Educational Computing Conference, Chicago, IL, United States. (ERIC Document Reproduction Service No. D462949). https://files.eric.ed.gov/fulltext/ED462949.pdf

Stager, G. (2005, August 28–31). *Papertian constructionism and the design of productive contexts for learning* [Paper presentation]. EuroLogo X, Warsaw, Poland.

Stager, G. (2010). *A constructionist approach to teaching with robotics* [Conference presentation]. Constructionism 2010, Paris, France. http://stager.org/articles/stagerconstructionism2010.pdf

Stroh, D. P. (2015). *Systems thinking for social change.* Chelsea Green Publishing.

Tissenbaum, M., & Ottenbreit-Leftwich, A. (2020). A vision of K–12 computer science education for 2030. *Communications of the ACM, 63*(5), 42–44. https://doi.org/10.1145/3386910

Thompson, C. (2019, February 13). The secret history of women in coding. *The New York Times Magazine*. https://www.nytimes.com/2019/02/13/magazine/womencodingcomputer-programming.html

Tonbuloglu, B., & Tonbuloglu, I. (2019). The effect of unplugged coding activities on computational thinking skills of middle school students*. Informatics in Education, 18*(2), 403–426. https://doi.org/10.15388/infedu.2019.19

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research, 57*(1), 3–31. https://doi.org/10.1177/0735633117743918

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). *A model curriculum for K–12 computer science: Final report of the ACM K–12 Task Force Curriculum Committee.* Computer Science Teachers Association. https://dl.acm.org/doi/book/10.1145/2593247

Tuomi, P., Multisilta, J., Saarikoski, P., & Suominen, J. (2018). Coding skills as a success factor for a society. *Education and Information Technologies, 23*(1), 419–434. https://doi.org/10.1007/s10639-017-9611-4

Turan, S., & Aydoğdu, F. (2020). Effect of coding and robotic education on pre-school children's skills of scientific process. *Education and Information Technologies*, *25*, 4353–4363. https://doi.org/10.1007/s10639-020-10178-4

United States Bureau of Labor Statistics. (2020). *Computer and information technology occupations*. https://www.bls.gov/ooh/computer-and-information-technology/home.htm

United States Department of Education, Office of Educational Technology. (2016). *Future ready learning: Reimagining the role of technology in education.* http://tech.ed.gov/files2015/12/NETP16.pdf

Uzunboylu, H., Kınık, E., & Kanbul, S. (2017). An analysis of countries which have integrated coding into their curricula and the content analysis of academic studies on coding training in Turkey. *TEM Journal*, *6*(4), 783–791. https://doi.org/10.18421/TEM64-18

Vegas, E. & Fowler, B. (2020, August 4). *What do we know about the expansion of K–12 computer science education?* The Brookings Institution. https://www.brookings.edu/research/what-do-we-know-about-the-expansion-of-k-12-computer-science-education/

Wang, J., Hong, H., Ravitz, J., & Moghadam, S. J. (2016, February). Landscape of K–12 computer science education in the U.S.: Perceptions, access, and barriers. SIGCSE '16: Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, TN, United States.

Wang, J., & Moghadam, S. J. (2017, March). *Diversity barriers in K–12 computer science education: Structural and social*. SIGCSE '17: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. Seattle, WA, United States.

Weissmann, G. S., Ibarra, R. A., Howland-Davis, M., & Lammey, M. V. (2019). The multicontext path to redefining how we access and think about diversity, equity, and inclusion in STEM. *Journal of Geoscience Education, 67*(4), 320–329. https://doi.org/10.1080/10899995.2019.1620527

Wenglinsky, H. (2005). Technology and achievement: The bottom line. *Educational Technology, 63*(4). https://imoberg.com/files/Technology_and_Achievement_--_The_Bottom_Line_Wenglinsky_H._.pdf

Williams, H. E., Williams, S., & Kendall, K. (2020, June). *CS in schools: Developing a sustainable coding programme in Australian schools* [Paper presentation]. ITiCSE '20: Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, Trondheim, Norway. https://dl.acm.org/doi/pdf/10.1145/3341525.3387422

Wilson, A., & Moffat, D. C. (2010, January 1). *Evaluating Scratch to introduce younger schoolchildren to Programming* [Paper presentation]. Psychology of Programming Interest Group 2010 Workshop, Madrid, Spain. https://scratched.gse.harvard.edu/sites/default/files/wilson-moffat-ppig2010-final.pdf

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A, 366*(1181), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wong, G. K. W., Ching, E., Mark, K. P., Tang, J. K. T., Lei, C. U., Cheung, H. Y., & Chui, M. (2015, January 3–5). *Impact of computational thinking through coding in K–12 education: A pilot study in Hong Kong* [Conference presentation]. Proceedings of the 11th International Conference on Technology Education in the Asia Pacific Region, Hong Kong. https://www.researchgate.net/publication/272122337

Wong, G. K. W., & Jiang, S. (2018). Computational thinking education for children: Algorithmic thinking and debugging. *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. https://doi.org/10.1109/TALE.2018.8615232

World Economic Forum. (2015). *New vision for education: Unlocking the potential of technology.* https://widgets.weforum.org/nve-2015/index.html

Wu, L., Looi, C. K., Multisilta, J., How, M. L., Choi, H., Hsu, T. C., & Tuomi, P. (2020). Teacher's perceptions and readiness to teach coding skills: A comparative study between Finland, mainland China, Singapore, Taiwan, and South Korea. *Asian-Pacific Education Research, 29*(1), 21–34. https://doi.org/10.1007/s40299-019-00485-x

Yadav, A., Gretter, S. Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235–254. https://doi.org/10.1080/08993408.2016.1257418