COMPUTER PROGRAMMING/CODING, ROBOTICS AND LITERACY: A QUALITATIVE CONTENT ANALYSIS

Dissertation

Presented to

The Faculty of the Department Language, Literacy and Special Populations

Sam Houston State University

In Partial Fulfillment

of the Requirements for the Degree of

Doctor of Literacy

---

by

Renee Elizabeth O’Neal

December, 2019

COMPUTER PROGRAMMING/CODING, ROBOTICS AND LITERACY: A

QUALITATIVE CONTENT ANALYSIS

by

Renee Elizabeth O'Neal

____

APPROVED:

Debra Price, PhD
Committee Director

Patricia Durham, PhD
Committee Co-Director

Nancy Votteler, PhD
Committee Member

Tonia Dousay, PhD
Committee Member

Stacey Edmonson, PhD
Dean, College of Education

DEDICATION

When I started working on this degree, my mom had just found out that her leukemia was in remission. I often visualized pushing her wheelchair into my defense and her being there when I was officially announced as a "Doctor." After the completion of my second year of classes, my mom unexpectedly passed away. It was a heartfelt dream that she would be there when I finished.

I dedicate this degree to my Mom. The one that fought through so much of her life, constantly dealing with illness and pain. She was a fighter and the most stubborn and determined woman I have ever known. I feel like because of her I was able to continue and see this degree to completion.

All of the events that you are told should not happen while working on your doctorate, happened to me. My mother passed away. I changed my job, and I had a major surgery. Through it all, I could always count on Dr. Price and Dr. Votteler to be there and help me keep it all together. Dr. Votteler would always say that she was the "lion in my corner." When I didn't know how to get started after mom passed away, it was Dr. Price that called me in to meet with her and had a plan ready for me. She not only showed me where to start, but she showed me how to get where I needed to go, and who would help me get there. That was Dr. Durham. Dr. Durham guided me every step of the way, helped me set and meet timelines, and see this degree to completion.

At home, I spent many, many hours away from my family upstairs writing. I also dedicate this degree to them, my husband Eric, and my three daughters Morgan, Haylie and Mackenzie. They have supported me every step of the way, even when I couldn't see where to go next.

ABSTRACT

O'Neal, Renee E., *Computer Programing/Coding, Robotics and Literacy: A Qualitative Content Analysis.* Doctorate in Reading (Literacy), September 2019, Sam Houston State University, Huntsville, Texas.

The purpose of this study was to determine if any theories emerged when examining the steps of learning to code/program on the computer and the developmental stages of learning to read and write. This was a Qualitative Content Analysis, Grounded Theory study. The three data sources included scholarly journals, various coding standards required by states in the United States, and websites that teach individuals how to code/program. Three cycles of coding were completed with each data source and the process was repeated with the outcome results from the three data sources inclusively.

The theories that emerged proved that there are similar skills that develop as a student is learning to code/program and learning to read and write. Those skills include sequencing, inferencing, problem solving, computational thinking and communication skills. The implications for this study will lead to more research in the field of coding/programming and the movement to include it in state required standards.
KEY WORDS: Coding, Programming, Reading, Writing, K-12th grades, Grounded Theory, Qualitative Content Analysis.

ACKNOWLEDGEMENTS

I would like to acknowledge my entire dissertation committee, Dr. Debbie Price, Dr. Patricia Durham, Dr. Nancy Votteler, and Dr. Tonia Dousay. Through this entire process they have been there for me to ask questions, figure out the next steps, and cry on their shoulders.

In addition, I would like to acknowledge Dr. Hannah Gerber. Dr. Gerber lead me in the direction of coding and digital literacy. Without her, I might still be searching for a topic to research. She also introduced me to many research leaders in the field of digital literacy that contributed to this dissertation. And also, my very dear friend Dr. Tara Wilson, who opened my eyes to the possibility of pursuing this degree in the first place.

Dr. Ronny Knox, for supporting me during this new job and allowing me "writing days." Another strong influence during this process was Dr. Susan Borg. While I was attempting to write this dissertation, I was also working on my Superintendent certification. She worked with me to ensure that I was able to complete both goals of graduating and finishing my Superintendent certification.

I would also like to acknowledge my husband and say thank you for all of your support in everything I strive to achieve. Through everything you are always there for me, and I do not know what I would do without you.

Morgan, Haylie and Mackenzie, my three beautiful daughters always remember you can achieve anything in this world. Don't ever let anyone tell you otherwise.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

CHAPTER I

Introduction

"Literacy is a lot of things. Its metaphors never tell the whole story, but by allowing us to take a particular angle into literacy, they show us glimpses of the ways communications happen in a society."

Sylvia Scribner (1984, p. 6)

Background

Our current society has become dependent on the use of technology. According to the *STEM Occupations: Past, Present and Future Spotlight* (2017) published by the United States Bureau of labor statistics, employment in science, technology, engineering, and mathematics (STEM) occupations grew by 10.5% or 817,260 jobs between 2009 and 2015, compared with slightly more than five percent net growth in non-STEM occupations (Fayer, Lacey, & Watson, 2017). Employment in computer occupations approached 3.2 million in May 2009 and nearly 3.9 million in May 2015 (Fayer, Lacey, & Watson, 2017). The group of occupational computer job openings is projected to yield over 1 million job openings from 2014 to 2024. Additionally, between May 2009 and May 2015, states in the United States that added the largest number of STEM jobs included California (160,950), Texas (102, 190), New York (42,980), and Michigan (41,100) (Fayer, Lacey, & Watson, 2017). Computer programming/coding has increasingly become a job of the future, and students should be exposed to learning how to program prior to entering college level studies.

Computer coding uses a variety of terms interchangeably including coding, computing, scripting, programming, and/or computational thinking. Across the globe, many countries are now requiring educators to teach coding beginning in the elementary grades.

England began requiring coding starting in kindergarten in the school year 2014-2015 (Furber, 2012). As part of its new digital technologies' standard, Australia has included programming starting in second grade (ACARA, 2016). By 2020, over a dozen European countries will require that computing will be a compulsory part of the curriculum (Engelhardt & Balanskat, 2015). Countries with more decentralized education systems such as Spain (Valverde-Berrocoso, Fernandez-Sanchez, & Garrido-Arroy, 2015), Germany (Deckler & Ifenthaler, 2017), and the United States (CS4RI, 2016; Riberiro, 2013; Smith, 2015) are trending as well in requiring teachers to include learning to code in the state standards. Gardiner (2014), reported research from Estonia where first graders were learning how to code and create their own computer games. In 2008, Pete Lomas, Managing Director of Norcott Technologies and David Braben the co-author of the seminal BBC micro game Elite formed the Raspberry Pi Foundation (raspberrypi.org, 2009). It was through this collaboration that the team created the mission to allow students to be taught how to make a computer "do something" instead of using the computer as a reward activity after they finish their regular curriculum work (raspberrypi.org, 2009).

Digitalization, identified as a major way to increase productivity in the public sector job market is needed in order to stay relevant in the innovative programming jobs of the future (Tuomi, Multisilta, Saarikoski & Suominen, 2018). According to experts in the field of programming and coding, digital literacy is an essential component of a modern-day education (Gardiner, 2014; Lohr, 2015). Coding, programming, computational thinking involves learning to think, represent, and solve problems that require a computation of human cognitive power, and computing capacity (Computer Science Telecommunications Board [CSTB], 2010; Kafai & Burke, 2014; Lye & Koh, 2014; Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013).

During the 1990s teachers, school administrators, parents, and community members began recognizing technology as a progressive educational tool that was a necessity in American education and the competitive global marketplaces of the 21st century. In 1965, President Johnson signed the Elementary and Secondary Education Act, a civil rights act that provided education funding to states and attempted to ensure that every student had access to an education (McLaughlin, 1974; Selfe, 1999).

The Clinton administration established the Goals 2000: Educate America Act HR 1804 (1994), as a response to the public's increasing dissatisfaction with the education system. The Goals 2000 Act included expectations for each state to set forth an agenda to: (a) develop comprehensive strategies for helping all students reach challenging academic standards, (b) revise and upgrade both curriculum and assessments, (c) improve teaching, (d) strengthen the instructional accountability, and, (e) expand technology efforts in educational settings. In 2002, Congress passed the No Child Left Behind Act (NCLB, 2001) as a reauthorization of the Elementary and Secondary Education Act of 1965. President Bush signed the NCLB into law in January 2002. In terms of technology, NCLB required the improvement of student achievement in elementary and secondary schools through integration initiatives, building access, accessibility, and parental involvement. Additionally, the NCLB Act emphasized the integration of technology for principals, teachers, and other school staff, including training for implementation for all instructional staff. The NCLB Act goals for Enhancing Education through Technology Part D included the following:

1. To improve student academic achievement through the use of technology in elementary and secondary schools.
2. To assist every student in crossing the digital divide by ensuring that every student is technologically literate by the time the student finishes eighth grade, regardless of the student's race, ethnicity, gender, family income, geographic location, or disability.
3. To encourage the effective integration of technology resources and systems with teacher training and curriculum development to establish research-based instructional methods that can be implemented as best practices by state and local agencies.

During the Obama administration, Every Student Succeeds Act (2015) was mandated leaving educators frustrated by the overabundance of testing and assessments that were part of the No Child Left Behind Act (2001). President Obama stated that the NCLB Act did not consider the individual needs of each community and resulted in a cookie cutter type of education: one size fits all. The primary goal of technology within the NCLB Act was to improve the student academic achievement through the use of technology in elementary and secondary schools (www.2.ed.gov). In addition, NCLB strived to assist districts with allowing every student to have access to technology with the goal of every student being technologically literate by the time the student finished the eighth grade, regardless of the student's race, ethnicity, gender, family income, geographic location, or disability (NCLB, 2001). Another expectation of NCLB was to encourage the effective integration of technology resources and systems to establish research-based instructional methods that can be widely implemented as best practices by State educational agencies and local educational agencies, as well as teacher training and curriculum development (NCLB, 2001).

Every Student Succeeds Act (ESSA), enacted in December 2015 governed K-12 public education in the United States. The law replaced its predecessor, the NCLB Act, and modified, but did not eliminate, provisions relating to the periodic standardized tests administered to students. Like NCLB, ESSA is a reauthorization of the Elementary and Secondary Education Act of 1965, which established the federal government's expanded role in public education.

The ESSA leaves significantly more control to the states and districts in determining the standards students must meet. States are required however, to submit their goals and standards to the United States Department of Education (DOE). The DOE submits feedback which all of the states must take into consideration. After revising, each state returns the updated documents, and the DOE approves them. According to ESSA accountability measures, the DOE still holds states accountable by ensuring they are implementing complete and ambitious, yet feasible goals.

With the implementation of ESSA, for the first time, American school districts faced the requirement to teach all students to high academic standards that will directly prepare them for success in college and career choices. (ESSA, 2015). ESSA accelerated the movement of Universal Design for Learning (understood.org). Universal Design for Learning (UDL) is an expectation of ESSA which incorporates offering information to students in more than one format, for example, through technology avenues. In offering students more access to content through technology, UDL allows students who learn in different modalities to access more of the curriculum. The incorporation of UDL as required by ESSA, gives students more than one way to interact with learning material and presenting what they have learned through technology. Assistive technology devices and access for students with special needs is also a critical compliance component of ESSA (ESSA, 2015).

When President Trump took office, he promised to make education a priority and to support education endeavors. The five major themes in President Trump's 2018 Budget are as follows: (a) expanding school choice, ensuring more children have equitable opportunities to receive quality education; (b) maintaining strong support for the Nation's most vulnerable students; (c) simplifying funding for postsecondary education; (d) continuing to build evidence around educational innovation; (e) eliminating or reducing DOE programs consistent with the limited Federal role in education. (President Trumps' FY 2018 Budget, 2018) (p. 1)

…Betsy DeVos, the Secretary of Education under President Trump's administration, and the United States Department of Education developed a Strategic Plan for Fiscal years 2018-2022 to implement the changes required by ESSA. College and career readiness standards, innovated ways of learning, and school choice for learning environments are three key elements of ESSA. Specifically, strategic objective 1.3 which is to prepare all students for successful transition into college and careers by supporting multiple options to expand access to high-quality science, technology, engineering, and mathematics (STEM), (United States Strategic plan for Fiscal years 2018-2020). In addition, strategic objective 2.2, requires partnerships with high demand industries to expose students to skills necessary to provide seamless pathways to high demand jobs in computer coding/programming, and other STEM fields (United States Strategic plan for Fiscal years 2018-2020).

With the congressional and presidential authorization of the Goals 2000 Act in 1994, the NCLB Act of 2001, and the ESSA of 2015, technology was mandated to be an important curriculum component. In recent legislation technology, computer science, coding, robotics, and engineering curriculum have been incorporated into state expectations for school districts. The definition of what it means to be literate in society has evolved over time in response to the technological advances.

Statement of the Problem

Over the past 20 years, the term "being literate" has evolved. Many new literacies have emerged, specifically digital literacies. Historically, literacy was synonymous with being a successful, productive citizen (Smith, 2002). Skills that are considered literacies and reflect a well-rounded, educated individual in society include: (a) health literacy, (b) financial literacy, (c) cultural literacy, (d) visual literacy, (e) technological literacy, and (f) the original definition of literacy, the ability to read and write (Smith, 2002). To be successful in the present-day economic society, individuals should extend that literacy knowledge to computer programming and coding.

What is considered literacy can proliferate real changes in skills required of students, workers, and everyday citizens, as well as changes in perceptions of what is to be considered a productive worker or citizen. According to recent literature, the skill of coding is becoming more and more a part of literacy. Terms such as non-coder or nonprogrammer have begun to emerge, and software is being created for "coding for the non-coder" or "installation for the non-coder" (Vee, 2013, 2017).

Originally, literacy began as a religious virtue, a way for individuals to connect directly with God (Smith, 2002). As the government began to sponsor education in the 19th century, literacy became a civic rather than a religious virtue (Smith, 2002). During the World War II era, computers were initially being developed and the demands on soldiers, nations and citizens began to be more complex, and the technologies through which literacy impacted began to figure into literacy's valuation (Papert, 1993). At this point in the history of our society, reading, writing, and programming literacy began to emerge as skills to learn in order to be a productive citizen. (Vee, 2017).

Through compulsory schooling, which promoted literacy, the Industrial Revolution in 19^{th} century Europe and the United States became the economic, and social order. Moving into the 20th century, education was available to the masses. The inability to achieve literacy was seen as an individual failure and indicative of other character weaknesses.

Information processing for war time supported literacy initiatives in the United States and also led to research in computation. As computer technology and programming developed during World War II, the rhetoric for programming as a new literacy began to develop. The first effort to teach programming came from Kemeny and Kurtz's National Science Foundation Grant (1974), funded by the drive to teach programming to all Dartmouth undergrads in the early 1960s. Kemeny and Kurtz argued that future leaders needed programming to understand modern systems of communication (1974). Forsythe (1959) of Stanford University Mathematics Department wrote:

> …we think every undergraduate mathematics student should know how to code some machine fairly well. (I would also include all undergraduate students, for I feel that the computer revolution will have such a great impact on all of our lives that ever college graduate should understand it ultimately. Possibly it will eventually be taught in the ninth grade for the same reason). Since coding presupposes no mathematics beyond arithmetic, it can be taught to freshman. I recommend a two-hour-per-week semester course in coding, to be taken as early as possible (Forsythe, 1959).

Another early advocate of including coding and computers in school was Alan Perlis. Perlis in 1960 directed the computation center at Carnegie Tech. His research was focused primarily on developing programming language and introducing programming coursework. At "The Use of Computers in Engineering Classroom Instruction," conference in 1960, Perlis commented that computers were tools of formal reasoning and should be available to all students upon entering the university. At a 1961 MIT conference titled, "Computers and the World of the Future," Perlis reiterated this vision.

Purpose of the Research Project

The primary purpose of this study was to determine the theories that emerged between the topics of learning to code and learning to read and write. The secondary purpose of this research project was to illuminate the literacy of computer coding, and the impact of learning to code on reading and writing skills. Educators and programmers have previously made the connection between programming and literacy. The theories that shaped and continue to shape coding literacy were also researched. I discovered theories of how learning to code aligned with the stages of learning to read and write. There have been many previous studies on how learning to code can improve science, computer science, and STEM related skills and interests in students (Bertran, 2016; Fayer, Lacey, & Watson, 2017; Glister, 1997; McDonald, 2016). There is a gap in the research connecting coding literacy to reading and writing literacy. According to Vee, (2017) to lack the knowledge of something considered literacy, is to be illiterate. Computer code is layered within the technology of writing and structures much of our modern-day communications, including word processing, e-mail, the Internet, social network platforms, digital video production, and mobile phone technology. The terms coding and programming will be used interchangeably throughout this study.

Programming and writing have a complicated association. Programming is considered a form of writing because symbols are used to create a message to read. Although programming is more than writing, it is a planned sequence of symbols that will be read, executed, or run, by the computer (DiSessa, 2001; Haas, 2009; Holdstein & Selfe, 1990; Vee, 2017).

The purpose and focus of studying coding in the elementary school was to determine the impact that learning to code can have on literacy skills, specifically developmental reading and writing skills. New media literacies are the social-cultural skills necessary for navigating our current world. No longer is information only conveyed by lines on paper that make up words, we also receive information through images, sounds, and multimedia representations. In the 21st century, children need to be fluent in reading, understanding, and communicating with different forms of multimedia. Section 9.3 of the United States Government's National Broadband Plan (Federal Communications Commission, 2010) stated that no universal definition of digital literacy exists because the definition continues to evolve. Technology, and how we use technology, is evolving daily. In schools today, the phrase "computer-aided instruction" means making the computer help teach the child. In Papert's (1993) vision, the child programs the computer and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, mathematics, and the art of intellectual model building. Programming a computer means nothing more than communicating to it in a language that it and the human user can both understand. Almost all children learn to talk, why should a child not learn to talk to a computer and make a computer "do" what the child programs it to do (American Library Association [ALA], 2000; Jenkins, 2008; Papert, 1993; Thoman & Jolls, 2004).

The Problem and Questions

Within our surroundings, we encounter digital technologies daily. Some of the typical technology experiences we encounter each day include: (a) electronic paper towel holders, (b) cell phones, (c) GPS devices, (d) electronic cars, gasoline pumps, and (e) credit/debit card machines. In the late 1970s, Apple Computer introduced the Apple II, which was a Central Processing Unit (CPU) that included a 4kb standard memory and customers used their televisions as monitors (Apple Technology Company, 1976). At that time, there were approximately 200,000 personal computers in the world. Today, over 200,000 personal computers are manufactured in a single week, playing a central role in our personal and professional lives, and in educational environments.

Papert (1993) was among the first to see that a massive change was needed in the education system, particularly in math and science. He was also one of the first to recognize that technology in the classroom was not a singular solution that would solve all of education's problems, real or perceived. Papert (1993) further stated that technology in education is effective only if placed in a larger context that combines well-prepared teachers with an integrated curriculum. Papert (1993) also recognized the potential for a computer to fundamentally transform the way people think, work, learn, and communicate.

The group of entrepreneurs who created Apple Computer acknowledged the intuitive ability of children between the ages of one and three to absorb information in massive amounts that is unequaled at any other time in their lives (Apple Technology Company, 1976). Also, this group believed that a child can learn two languages without confusing the two. John Scully, Chairman and CEO of Apple Computer, Inc. proports that the thoughtful introduction of information technology into a social system can fundamentally transform the way intelligence is developed and the manner in which people live their lives. According to Papert, it is rare that an exceptional event leads people to reorganize their intellectual self-image in such a way as to open up new perspectives on what is learnable (1993).

Within this study, I examined three data sources: scholarly articles, coding standards which included state and national scope and sequence guides, and websites that allow students to learn to code. The questions that I sought to answer in this study were:

1. What theory emerged that connected learning to code and coding literacy to the developmental stages of learning to read and write?
2. What are the theories that are shaping coding literacy?
3. Does the current data on learning to code and learning to read and write intersect and what theories emerged from the two?

Definition of Terms

Coding- The act of informing the hard drive of a computer with the language that will create the actions within a computer's hard drive that you are desiring to implement.

Digital literacy-All digital literacies have in common the aspect of computer programming.

Computational thinking- Critical thinking involves problem analysis and decomposition, algorithmic thinking, algorithmic expression, abstraction and modeling.

Literacy- Literacy means to attach that skill with the moral weight and importance of reading and writing.

Programming languages- Programming languages are a formal language that specifies a set of instructions that can be used to produce various kinds of output. Programming languages can be used to create programs that implement specific algorithms within a computer.

Computer science- Computer science is the study of computers.

Robotics in education- A curriculum that allows students to learn how to build and program their own robot.

Digital technologies- This is the branch of scientific knowledge that deals with the creation and practical use of digital or computerized devices, methods, and systems.

Stages of reading development. According to Chall (1976), the five stages of reading development are:

Stage 0 (up to age 6) is a prereading stage that is characterized by children's growth in knowledge and use of spoken language.

Stage 1 (Grades 1-2), children learn the letters of the alphabet and the correspondences between letters and the sounds that they represent.

Stage 2 (Grades 2-3), children confirm what is learned in Stage 1 and learn to apply the knowledge gained in Stage 1 to read words and stories.

Stage 1 and 2 together, constitute a learning to read stage at the end of which children recognize most words automatically and read passages with ease and expression.

Stage 3 (Phase A, Grades 4-6; Phase B, Grades 7-8 and/or 9), students begin to learn new knowledge, information, thoughts, and experiences by reading.

Stage 4 (high school), students learn to deal with multiple viewpoints. What the students are reading incorporates more than one set of facts, competing theories, and multiple interpretations.

Stage 5 (age 18 and above), readers can read materials in the degree of detail and completeness that is needed to serve their purposes.

Stages of writing development. According to Graves (1994), the five stages of writing development are

Stage 1. Spelling

Stage 2. Motor Aesthetic

Stage 3. Conventions

Stage 4. Topic information

Stage 5. Revision.

Significance of the Problem and the Justification for Investigation

Technology is everywhere in today's society. More and more companies are going paperless and conducting all business communications and transactions electronically. By 2024, there will be approximately 4.4 million jobs available in the field of computers and information technologies (newamerica.org, 1999). The first mention of integrating programming and coding into the school curriculum was inspired by Papert (1980) in his book, *Mindstorms: Children, Computers and Powerful ideas*. He argued that learning to code and program provided children with an opportunity to think about their own thinking. At the time, the book was controversial, but it helped educators understand the importance of having computers in schools. The theories that emerge from this study will contribute to the success of our future nation. As coding becomes a necessary life skill/literacy, it is imperative that we provide all students with the ability to code. And, if we can align teaching coding with the research that exists on learning to read and write, computer coding/programming will become a pedagogy that is easily accessible to the educators who, teach our children and a critical part of the standards that are taught in schools.

Summary

"We are on the verge of new technological revolutions that could improve, virtually, every aspect of our lives, create vast new wealth for American workers and families, and open up bold, new frontiers in science, medicine and communication."

-*President Donald J. Trump, (2018)*

Coding, programming, and digital literacies are critical to the future of our country. In order to make our children college and career ready, they must have knowledge of digital literacies. This chapter explains the purpose of this study which is to determine the theories that emerged between the topics of learning to code and learning to read and write. In the next chapter, I provided the reader with a comprehensive, informative review of literature on coding literacy and its connection to learning to read and write.

CHAPTER II

Review of Literature

As technology and the ways that it impacts our world continues to develop, it is important to look at methods to change our school curriculums and prepare our students for future careers and productive citizenship. The investigative topic of computer coding/programming and how it is synonymous with learning how to read and write is important.

Looking at programming from the perspective of literacy and literacy from the prospective of programming, I make two central arguments: (1) programming shows us what literacy looks like in a moment of profound change; (2) the history and practices of reading and writing human languages can provide useful comparative contexts for contemporary programming. (Vee, 2017). As educators, in order to stay current and relevant we must stay up to date with the most current research and effective teaching strategies to provide the best education program possible for the future leaders and citizens of our country.

Throughout the research process for this literature review, I read to discover emerging theories from which learning to code and learning to read and write can intersect. I searched for commonalities across the coding, and reading, and writing disciplines. As a reader of this study, I presented the common theories that emerged from examining the processes for learning to code and learning to read and write. I examined current state and national coding standards to determine if there was a correlation between the learning expectations for reading and writing literacy and coding as a literacy. As I researched databases, I included the following key words; reading, writing, coding, programming, and robotics.

As presented in Chapter 1, being literate is often considered synonymous with being successful and productive in society. Programming is considered a form of writing, because conceptually, programming consists of symbols inscribed on a surface and created to be read and executed by a computer. When programming moved from physical wiring and direct representation of electro-mechanics to a system of symbolic representation, it became a form of writing. Recent changes in programming language design, such as AppleScript, lend themselves towards words rather than numbers: language libraries, code comments, automatic memory management, and structured program organization which are all elements of programming language that enhance the legibility of code. We now write in a language that computers can understand (Nofre, Priestley & Alberts, 2014).“Literacy reflects real knowledge requirements (its functional requirement), as well as the perception of what kind of knowledge is required to get around in society (rhetorical component)” (Vee, 2017, p. 5).

Scribner (1984) describes literacy as three major metaphors. The first is functional literacy, which refers to the minimum literacy deemed necessary for successful social and economic participation in society; the ability to read and write ideas and needs. The second metaphor from Scribner is literacy as power. The power of literacy is revealed by the fact that education is designed and tends to promote the kinds of literacy possessed by higher echelons of society. Thirdly, literacy is a state of grace that is simultaneously adaptive, socially empowering, and self-enhancing by allowing access to the world and its contents.

Papert (1980), who studied under Piaget, designed the Logo programming language. The intent was to scaffold students into learning complex logic, physics, and problem solving through programming. Similar to others who are considered builders, children will appropriate to their surroundings and incorporate the models and metaphors suggested by their surrounding culture. Society views literacy as a set of skills necessary to be productive and purposeful. These skills include: (a) health literacy (the ability to obtain, read understand, and use healthcare information in order to make appropriate health decisions) , (b) financial literacy (the ability to manage your money, pay bills, and how to borrow and save money), (c) cultural literacy (the ability to function and understand fluently in a given culture), and (d) digital literacy (the ability to use and apply information and communication technologies). To call something a literacy is to raise the stakes for acquiring that knowledge and conversely, to lack the knowledge of something considered a literacy is to be illiterate. Literacy as a standalone term generally implies reading/textual literacy. In our current world, computers are part of our infrastructure including communications, word processing, social networking, digital video production, and mobile phones. A skill can be referred to as a literacy when it acquires the moral weight and importance of reading and writing in our society. In 1994, Hamilton, Barton and Ivanic referred to illiteracy as a disease, a possible link to criminality, a drain on the economy, and a cause for joblessness and individuals being held back from reaching their full potential. For example, accounting tasks that used to be manual with physical books, creating paper-based duplicate copy invoices are now being composed digitally using specialized software. Similarly, writing a letter by hand to invite a new business partner to engage in a proposal might be accomplished by sending an email with digital attachments with responsive data reports based on a specific industry. The ability to manipulate the use of a computer, for work or personal use, is now a necessity of life.

As a citizen of Brazil, Freire commissioned to head the country's literacy campaign in the 1960s. Freire believed education could be a political act that could not be separate from pedagogy. While Freire is best known for his work on the pedagogy of the oppressed, he firmly taught that literacy was a tool of liberation. He also professed that literacy was a road to help people learn and remake themselves. Literacy provided power, intellectually and politically. Freire developed his own method of literacy training to address the ever-growing need for educating the poor in Brazil. Freire coined his literacy method as conscientization. The process of conscientization referred to the act of when individuals, as knowing subjects, achieve a deepening awareness of the socio-cultural reality which surrounds them in life, and of their ability to transform that reality through action. Literacy provides individuals access to jobs and a resource for living in everyday life (Bers, 2018; Freire, 2017).

Reimagining the Role of Technology in Education

If programming continues to be seen as a form of thinking and engaging the individual mind, and not as a form of participation and expression, then we will fail to benefit from learning and teaching code in today's networked communities. Understanding the basics of coding allows us to understand how code underlies the interfaces, technologies, and systems that surround our daily lives. The ability to code empowers people to have higher paying jobs in computer technology fields. Reading code is as much about reading the world as it is about understanding, changing, and continuing to advance the digital world in which we live. According to Dewey, when children learn through exploring and their own doing, thoughts and ideas, critical thinking takes place (1938). Dewey also suggests that students learn best by the use of what is current in their worlds. Coding and programming are relevant skills that are now included as part of being literate. By incorporating what is real and what is present, students become excited, interested, and motivated to learn (Kafai & Burke, 2014).

Learning to code does not only impact the skill of making a computer do what you tell it to do. Coding by nature integrates multiple interdisciplinary skills that are important for students to learn to be successful citizens. Gee (2013) argued for well-designed games that build literacy skills, create problem-solving spaces with feedback and clear outcomes that lead to real, deep, and consequential learning. Many types of digital activities provide authentic and motivating contexts for developing specialized vocabulary that will be needed in the future. Coding introduces students to the disciplinary literacies of computer engineering, game designers, graphic designers, and much more. Interacting with coding apps allows students 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. According to the U. S. Bureau of Labor Statistics (BLS) biennial update of employment projections, over the next ten years, the demand for jobs that require the knowledge of coding will increase by 17% from 2014 to 2024 (Fayer, Lacey & Watson, 2017). Coding apps are designed to encourage the creation of digital content rather than just consume it. The apps provide an avenue where students must think creatively, reason systematically, and work collaboratively to code.

In the Common Core State Standards (CCSS), which are implemented in the majority of the states in the United States, there are many standards that can be addressed through integrating coding into the curriculum. The act of coding employs a wide range of English Language Arts standards, including the following anchor standards: (1) CCSS.ELA-LITERACY.CCRA.W.4:

"Produce clear and coherent writing in which the development organization, and style are appropriate to task, purpose, and audience" (p.21) ; (2) CCSS.ELA-LITERACY.CCRA.SL.1: "Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building others' ideas and expressing their own clearly and persuasively" (p. 23); and (3) CCSS.ELA-LITERACY.CCRA.L.6: "Acquire and use accurately a range of general academic and domain-specific words and phrases sufficient for reading, writing, speaking and listening at the college and career readiness level." (p. 25).

Learning to code also supports students' understanding of if-then relationships, learning how to navigate informational text, and experiencing the importance of sequencing events correctly. Most coding apps used by students are designed to require a visual programming language that connects code blocks together to create commands. In order to initiate these commands, the student is required to navigate informational text that is presented as images and icons, written text, audio and video which requires users to learn domain-specific vocabulary. Logic, reasoning, and problem solving to create or manipulate digital content through the use of computational thinking are also skills applied by students through coding apps.

Effects of Technology on Literacy Skills

In our everyday lives, practically every aspect of our day is impacted by the ability, or inability to read and write. Almost everyone looks at a computer screen each day, a mobile phone screen, and a thousand other devices that require reading of symbols and letters. To order at a fast food restaurant, you must be able to read the menu; to deposit or withdraw money at a bank, you must know how to complete bank paperwork; to fill out paperwork at a doctor's office in order for him to provide appropriate medical care, you must be able to complete the patient documents; to drive an automobile you must be able to read the road signs. All of these skills require the ability to read and write.

According to diSessa (2001), the bases for literacy are organized into intricately structured subsystems with particular rules of operation, basic symbol sets, patterns of combination, conventions and means of interpretation. All of these subsystems have specific character, power and reach, and they also have limitations. In thinking about those subsystems of language, computer technology offers a range of inscription forms, game interfaces, contemplative browsing, and simulations and calculations, to name a few. Considering all of these forms and the probability of more to come, it is inconceivable that our current conception of literacy would not include technology.

diSessa (2001) described literacy as a socially widespread patterned deployment of skills and capabilities in a context of material support to achieve valued intellectual ends. The constant evolution of technology provides continuous development for how, when, and where we engage in reading and writing. Consider the following digital literacy example: a certain kind of manual dexterity and hand-eye coordination became relevant with the invention of the typewriter keyboard. Prior to the invention of the computer mouse, the ability to type quickly and accurately was not valued or necessary because no need existed. As typewriters have evolved into computer keyboards, the skill has only increased in necessity and can be found in school library curricula across the United States.

Glister (1997) introduced and defined the term digital literacy. Glister explained that digital literacy is an ability to apply information from a variety of digital sources. He describes digital literacy as the ability to read, write, and interact with information from a variety of technologies. In Glister's 1997 book, *Digital Literacy*, he explained that to have digital literacy is to be able to acquire and master ideas from technologies, it is not just keystroking. The concept of digital literacy is very broad, it incorporates specific skills and competencies and the ability to generalize those skills across various technology outlets (Bawden, 2001).

Piaget (1977) described learning as the process of creating artifacts of personal and social relevance, connecting old knowledge to new knowledge, and interacting with others. Connected learning explores how the computer should be thought of as a tool for knowledge transformation, personal expression, and social relevance so that it allows learners to create and connect with others. The objects and images that we create in our minds do not exist in isolation from our real worlds and the communities that we experience every day. Thus, learning is how we process and learn new skills by attaching the new to previous learning (Kafai & Burke, 2014).

Similar to Piaget, Papert (1993) saw learning as building knowledge structures but he added another dimension. The images of the mind have to move into the real world where they can be examined, shared and valued by others. This added dimension is a predictor of the vast and distributed networks of digital objects and communities that comprise the internet. When children learn to write code, they learn to articulate procedures, recognize repetition, and *debug* their own thinking when a program does not run as expected. Learners can carry connected experiences, ideas and theories with them that are connected to personal experience. Papert (1993) described knowledge as the deliberate part of learning (that) consists of making connections between mental entities that already exist; new mental entities seem to come into existence in more subtle ways that escape conscious control…This suggests a strategy to facilitate learning by improving the connectivity in the learning environment, by actions on cultures rather than on individuals (p. 7). Thus, elementary children can learn programming and the act of coding can give them an instrument to develop and master skills in various required subject areas.

Clements (1999) visualized the future of educational computing research through the case of computer coding. He examined how computer programming can be infused into multidisciplinary subjects including mathematics, problem solving, higher order thinking, language arts, creativity, and social-emotional development (Clements, 1999). For this research, I focused on the language arts skills that Clements identified as having an impact on language skills (1999). In Clements' study, he determined that learning to program can increase a student's language skills (Lehrer & deBernard, 1987), readiness skills, visual discrimination, visual motor skills, visual memory (Reimer, 1985), and an increase in first graders scores on assessments of visual motor development, language, and listening comprehension (Robinson, Gilley, & Uhlig, 1988; Robinson & Uhlig, 1988). Learning to code, particularly in the Lego language, has been shown to increase reading skills, language mechanics, and reading comprehension (Studyvin & Moninger, 1986).

Educational theorists argue that gaming on the computer has created a new kind of literacy (Gee, 2003). In 2009, Owston, Wideman, Sinitskaya Ronda, and Brown purported that gaming combines significant elements that are also associated with reading and writing including accessing and evaluating information, constructing complex narratives, decision-making, and navigating digital environments. Also, in 2009, Turner & Katic completed a study on how high school students' writing processes were influenced by technology. They determined that as more students used technology for their writing, the more their fluency increased and the more they developed writing tasks by using this new literacy of technology (Turner & Katic, 2009).

Literacy and technology have been connected through various aspects. For example, a 2008 study investigated the effects of commercial electronic books for young children on kindergarteners' emergent literacy skills. Shamir, Korat, and Barbi measured multiple emergent literacy skills including: (a) phonological awareness, (b) reading words from an e-book, (c) word recognition, (d) story comprehension, and (e) story production. The results of the study suggested that the students who worked on e-book activities in a paired learning context saw larger gains in phonological awareness, emergent reading, and story comprehension over those who worked with it individually (Shamir, Korat, & Barbi, 2008).

Coding as a Literacy

Coding literacy is specifically the ability to give a computer symbolic commands in a specific order to produce a product on and through the computer hard drive. One of the co-founders of *Code.org*, Partovi, (2013) referenced that coding was the new 'American Dream' and should be available to everybody, not just the lucky few. The basic skills of reading and writing are undeniably required in order to master the skills required to code. The sequential order and patterns of letters are an integral part of reading as well as developing code (*Code.org*, 2013).

Advocates for computer technology have claimed that reading and writing code resembles textual literacy in many ways. Being literate allows people to represent their ideas in texts and to interpret texts produced by others. Reading and writing are technologies of textual literacy, while coding is a technology of computational literacy (Bers, 2018; Vee, 2013).

When thinking of coding as a literacy, an individual must produce an artifact or product by engaging in a process of creation. For us, coding is not a set of technical skills, but a new type of literacy and personal expression that is valuable for everyone, much like learning to write. According to Resnick and Siegel, coding is a new way for people to organize, express, and share their ideas (2015). An example would be Code Poetry, the phenomenon of intermixing notions of classical poetry and computer code. Through Code poetry, authors write poems in a programming language that are human readable as poetry. Additionally, Resnick and Siegel (2015) relate how coding is taught to traditional language arts instruction, stating:

In many introductory coding activities, students are asked to program the movements of a virtual character navigating through a set of obstacles toward a goal. This approach can help students learn some basic coding concepts, but it doesn't allow them to express themselves creatively—or develop a long-term engagement with coding. It's like offering a writing class that teaches only grammar and punctuation without providing students a change to write their own stories. (Resnick & Siegel, 2015)

Relationship between Coding and Learning to Read and Write

Judson (2009) argued that technology literacy gains could lead to heightened subject matter confidence and reflect improved ability to use technology as an avenue for new learning and application of that learning to other subject areas. The results of his study provided evidence of correlations between technology gains and literacy growth, using two theories to support the expectation that increased technology literacy leads to improved academic achievement. The first, confidence theory, supports the idea that improved technology literacy leads to heightened self-confidence and, in turn, fosters improved academic achievement (Anderman, Anderman, & Griesinger, 1999; Bandura, 1997). The second, mediation theory, purports that improved technology literacy represents an increased ability to use technology tools as mediators of knowledge (Vygotsky, 1978).

Pea and Kurland's (1984) seminal study into the cognitive effects of learning computer programming explored the frequently transitioning learning atmosphere in which our children experience new hardware and software daily. The researchers stated that they believe teaching students in a developmental approach to understand information technologies will be required in order to incorporate the new insights of cognitive science. This will contribute to future research and the design of computer-based learning environments. This developmental cognitive science discipline would merge theory and practice. Feurzeig, Bolt & Newman (1981) argued that teaching programming can provide the basis for the art of logical and rigorous thinking. According to Feurzeig and associates, learning to program is expected to bring about seven general changes in thinking:

1. Rigorous thinking, precise expression, the need to make assumptions;
2. Comprehension of general concepts such as formal procedure, variable, function, and transformation;
3. Greater facility with the art of "heuristics," such as finding a related problem and solving the problem by decomposing it into parts;
4. Constructing the notion that "debugging" of errors is a constructive activity;
5. Identification of the concept that one can invent small procedures to gradually construct solutions to larger problems;
6. Enhanced "self-consciousness and literacy about the process of problem solving";
7. Recognition that for areas beyond programming that there is rarely a single "best" way to do something; but respect different ways with respect to specific goals.

Programming is not a unitary skill. It is a complex skill that, like reading, is comprised of a combination of skills including memory and processing capacities, problem solving abilities such as comprehension, monitoring, inferencing, and hypothesis generation. The acquisition of programming urges us to develop artificial intelligence systems that "understand" natural language. Advanced reading requires wide experience with different genres (e.g. narrative, essays, poetry, and debate). There are also a variety of goals with reading, reading for content, reading for learning, and reading for voice and expression. To know how to read, you must know how to decode, which is similar to learning the language and vocabulary of programming. When thinking about reading as compared to programming, each context requires a specific set of skills, vocabulary, comprehension, and inferencing.

Incorporating programming activities of increasing cognitive complexity in school curriculums will promote the transfer of new programming knowledge to problem solving activities in other domains. Pea and Kurland (1984) described the three stages of programming skill development as: (a) Level 1, Program user; (b) Level 2, Code generator; (c) and Level 3, Program generator. The stages of programming skill development are much like a beginning reader, a developmental reader, and a fluent reader. According to Vee (2013), programming, like reading and writing, helps to develop social, technological, and cognitive skills. Most conclusively, programming, like writing have a direct correlation and process in order to be applicable.

Programming involves the act of writing code that tells the computer what to do. The programmer writes what is generally called the 'source code' and it is comprehensible to readers who understand the programming language. The next step is for the source code to be complied or interpreted by an intermediary program that translates the human-readable language to the language the computer can process. Programming is a form of writing because it is a set of symbols inscribed on a surface and designed to be read. Programming skills move beyond just writing, as the symbols are not only designed to be read but also to be executed, or run, by the computer. Presently, programming language includes the use of words, language libraries, code comments, automatic memory management, structured program organization, and the development of programming environments to enhance the legibility of code.

Specific to early childhood education, Kazakoff and Sullivan (2012) conducted a study on the effect of classroom-based intensive robotics and programming workshop on sequencing ability (2012). This study was conducted during a 1-week robotics workshop at an early childhood STEM school in the Harlem area of New York City (Kazakoff & Sullivan, 2012). The study found that there were significant increases in sequencing skills from pre-to-post tests.

Along with the changing in our technology world, the meaning of literacy has also changed. Children need to know much more than reading letters on a printed page, they also need to be fluent in understanding, and communicating with different forms of multimedia (American Library Association (ALA), 2000; Thoman & Jolls, 2004). The hypothesis of this study was that some of the processes involved in programming robots, specifically sequencing of programming commands, are also linked to the skills necessary to write and tell a story in a logical order (Kazakoff & Sullivan, 2012). The focus on robotics for this study involved utilizing robotics as a tool that can make abstract ideas more concrete because the students can directly and immediately see the effects of the programming steps in the actions of the robot (Kazakoff & Sullivan, 2012).

As earlier works in the field of emergent computer programming research were reviewed, the field suggests that children who participate in computer programming typically score around 16 points higher on cognitive-ability tests than children who did not participate (Liao & Bright, 1991). Specific skills noted in Kindergarten students who used the Logo programming language included sustained attention skills, independent self-direction, and higher rates of inquisitive thinking (Clements, 1987). More recent studies on innovative programming environments support the argument that children's programming of animations, graphical models, games and robots allows them to learn and apply core computational thinking skills such as automation, analysis, decomposition, modularization, abstraction, and iterative design (Bers & Horn, 2010; Mioduser, Levy & Talis, 2009; Mioduser & Levy, 2010; Resnick, 2006; Resnick et al., 2009).

Coding Standards

Included in this review are the computer science content standards for each age/grade level of students as dictated by state, local, and national initiatives and requirements. This data has been gathered, monitored, and updated to track the latest computer science/programming expectations and legislation by state through the cooperation of code.org. As of 2017, there has been an intense increase in interest amongst parents, teachers, school districts, and states to bring computer science and coding into our Kindergarten through 12th grade education system in the United States.

Table 1

Computer Science in K-12 schools

State	Funding	State plan	Secondary school requirement	Dedicated CS position	Implementation	K-12 CS certification
Alabama	X				X	
Arkansas	X	X	X	X		X
Arizona	X				X	X
California		X		X	X	X
Colorado	X				X	
Connecticut		X		X		
Delaware			X			
Florida			X	X	X	X
Georgia	X			X	X	X
Hawaii		X		X	X	
Idaho	X			X	X	X
Illinois					X	X
Indiana			X	X	X	
Iowa	X		X		X	X
Kentucky					X	X
Maine					X	
Maryland				X	X	X
Massachusetts	X	X		X	X	X
Michigan	X	X		X	X	X
Mississippi						X
New Hampshire		X		X	X	X
New Jersey			X		X	
New York	X	X			X	X
Nevada	X		X	X	X	X
North Carolina	X				X	X

Ohio					X	X
Oklahoma				X	X	X
Pennsylvania					X	
Rhode Island	X	X			X	
South Carolina					X	X
Texas					X	
Tennessee						X
Utah	X	X			X	X
Virginia	X			X	X	X
Washington	X			X	X	X
West Virginia			X		X	X
Wisconsin					X	X
Wyoming		X	X			X

**CS refers to Computer Science*

In Table 1, six subcategories indicate the stage states are currently at in establishing computer science integration. Almost all states have computer science standards developed, but only 16 provide funding for implementation. Additionally, 27 states have a computer science certification, but only 16 have a teacher whose only responsibility is to teach computer science and nine require computer science in secondary education to graduate from high school.

Summary

This chapter provided perspective on how programming entered our society and transformed the role of technology in education, technology affected literacy skills, and how the expectation for students to learn coding/programming entered our standards and curriculum spectrum. In the past, technology was viewed as something that could be added to the general curriculum, now it is an integral part. Not only do students need to learn to read and write as the traditional literacy, but students also need to learn coding literacy to be relevant in today's job market.

The next chapter provides a historical perspective of Grounded Theory and relevant methodology for conducting a Qualitative Content Analysis that explored the underlying issues related to learning to code and learning to read and write. I sought to answer the following questions:

1. What theory emerges connecting learning to code and coding literacy to the developmental stages of learning to read and write?
2. What are the theories that are shaping coding literacy?
3. Does the current data on learning to code and learning to read and write intersect and what theories emerge from the two?

The strands that run through the five categories described here in Chapter 2 were coded for data analysis in Chapter 4.

CHAPTER III

Methodology

An introduction to computer coding in elementary schools and related literature on robotics, literacy, and coding provided the foundation upon which the current project was framed. Additionally, a familiarity with the relationship between these fields, reading comprehension, and writing helps further conceptualize the process of learning how to code.

The following chapter provides an overview of the methodology of Qualitative Content Analysis using a secondary lens of Grounded Theory to better understand the messages and emerging theories on the reading and writing skills found in computer coding. Teaching curriculum that incorporates computer coding in elementary schools and the literacy skills that are acquired during the implementation were examined to uncover theory that influences curriculum writers. The following question guided this research:

> What theory emerges that connects learning to code and coding literacy to the developmental stages of learning to read and write?

This chapter includes the problem that was explored in this study and the questions that arose from the observations, analysis of existing literature, and themes in current research. The alignment of literacy skills and the implementation of coding standards were the focus of this study. A qualitative research approach provided an avenue to conduct a Qualitative Content Analysis through identifying categories and quantifying those categories. The data sources included examination of journal articles, databases and websites, and state and local computer science standards.

Methodological Framework

Qualitative Content Analysis. Qualitative Content Analysis is inductive and flows from a humanistic, not a positivistic tradition. Qualitative Content Analysis was first used to examine hymns, newspaper and magazine articles, advertisements and political speeches of the 19th century (White & Marsh, 2006). As a research method, Qualitative Content Analysis can be used to make replicable and valid inferences from data to their context, with the purpose of providing knowledge, new insights, a representation of facts, and a practical guide for future action. Krippendorff (2004) defined Qualitative Content Analysis as "a research technique for making replicable and valid inferences from texts to the contexts of their use" (p. 8). The research is limited in the area of computer coding and the connection to literacy development, therefore I proceeded with an inductive approach (Harwood & Garry, 2003; Krippendorff, 2004). According to Krippendorff (2004), "a Qualitative Content Analysis is a search for multiple interpretations by considering diverse voices (readers), alternative perspectives (from different ideological positions), oppositional readings (critiques), or varied uses of the texts examined (by different groups)" (p.18). Through this study, I examined and collected data on the contributions, or lack of contributions that learning to code could apply to literacy skills, specifically reading and writing. Bengtsson (2016) stated that "qualitative research contributes to an understanding of the human condition in different contexts and of a perceived situation" (p. 8).

Qualitative Content Analysis is a research method for analyzing, describing and interpreting texts (White & Marsh, 2006). In this study the text consisted of journal articles, standards, and websites. As I researched these text examples, I inductively read for significant phrases, clusters of meaning, and statements. A Qualitative Content Analysis can be described as a systematic approach to analyzing documents texts obtained in the course of research. Content Analysis has its background in the study of mass communications in the 1950's. In the years following, researchers in many fields including anthropology, library and information studies, political science, psychology, and sociology have also applied the method of Content Analysis. In Qualitative Content Analysis, the researcher reads and analyzes the text being researched to identify concepts, and patterns.

An example of a Qualitative Content Analysis in the field of literacy, was conducted by Marshall (2004). Marshall conducted a study examining the text in a selection of children's literature by bringing attention to the sex-role stereotypes. The researcher analyzed four variant versions of "The Little Red Riding Hood." The application of post structural feminist literary theory, through a text analysis, provided an avenue for the researcher to study the role of the female in various examples of "The Little Red Riding Hood" story.

Another example of applying Qualitative Content Analysis to a literacy study was conducted by Moss (2008). The purpose of her study was to compare the text genres in two state-adopted basal readers. Text samples were categorized into four genres: narrative, poetry, play, or nonfiction. As part of the text analysis, the researchers in this study computed the percentages of pages and selections identified from each of the genres in order to determine the amount of nonfiction text included in each.

Methodological Lens: Grounded Theory. The secondary methodological lens used for this study was Grounded Theory. Grounded Theory is defined by Glaser and

Strauss (1965) as the discovery of theory from data – systematically obtained and analyzed in social research. The key point in Grounded Theory is that the theory produced is grounded in the data. Grounded Theory was developed out of the need to generate new theories rather than force data into a few existing theories and the idea that qualitative and quantitative data are both useful (Urquhart, 2013). The rationale behind selecting Grounded Theory as a secondary lens was to complement Qualitative Content Analysis and allow for determination of the core variables in the data collection stage. Once the core variables were determined, recurring variables were extracted to analyze and summarize the findings. The core variable was allowed to emerge from the developing theory of systematic collection of data during this study. The Grounded Theory methodology is predominantly inductive, and I discerned what was directly related to the core variable. My goal was to generate emerging theory from the data that explained as much as possible, the elements related to connecting learning to code, and coding literacy to the developmental stages of learning to read and write (Urquhart, 2013).

While studying death and dying, Glaser and Strauss (1965) refocused qualitative inquiry on methods in analysis. During the early 1960s, Glaser and Strauss (1965) observed how dying occurred in a variety of hospital settings in the United States. They looked at how and when professionals and their terminal patients became aware that they were dying and how they handled the diagnosis. While they constructed their analysis of dying, Glaser and Strauss (1965) developed methodological strategies. In addition, Glaser and Strauss (1965) advocated developing theories from research grounded in qualitative data instead of deducing testable hypotheses from existing theories. The two researchers proposed that systematic qualitative analysis could generate theory (Glaser & Strauss, 1965). Referring to the development of Grounded Theory, Glaser (1998) stated, "A methodology was needed that could get through and beyond conjecture and preconception to exactly the underlying processes of what is going on so that professionals and laymen alike could intervene with confidence to help resolve the participants' main concern surrounding learning, pain and profit" (p.5).

Many have applied a Grounded Theory lens in educational settings. Webster and Son (2015) used semi-structured interviews, a survey questionnaire, and classroom observations to discover that the use of technology in the classroom exacerbates preexisting pedagogical and infrastructure issues, leading to inconsistencies in representation and application. Also, the authors revealed that utilizing technology in the classroom ceased because of the limitation of potential use by teachers. In another Grounded Theory study, the integration of an iPad app that focused on letter recognition in 6-7-year-old students incorporated a treatment group and control group (D'Agostino, Rodgers, Harmey, & Brownfield, 2015). The study also included a qualitative element examining teachers' perceptions of learning letter identification through the app. The study produced positive effects of the iPad app on the progress of students learning their letters. In addition, teachers noted a contradiction between their beliefs about literacy teaching, and learning and the app.

In a Grounded Theory study conducted in a Swedish school, Sward (2012) focused on upper level students re-learning to read and write in a forced individual program. To gather qualitative data, Sward (2012) used research interviews, observations, questionnaires, video recording, and analyses in the tradition of Grounded

Theory to discover that the teachers, who instructed in this forced reading and writing program for underperforming students, strived systematically to ensure every student's reading and writing development through didactic arranging; this means constant interaction with the individual student. Sward (2012) reported that the teachers in the study believed that all students could re-learn reading and writing, which helped the students believe in their (ability to re learn) re-learning and also to increase their self-esteem and self-confidence.

The secondary methodology lens of Grounded Theory was selected to uncover emerging theories, rather than forcing the data into an existing theory. The key idea is that the theory produced was grounded in the data and revealed insight into social relationships and behaviors. I discarded all previous theoretical ideas in order to let the theory evolve and emerge (Christiansen, 2007; Urquhart, 2013).

Research Questions

The following research questions were answered while applying the steps of Qualitative Content Analysis and determining the theories that emerged from the data.

1. What theory emerges that connects learning to code and coding literacy to the developmental stages of learning to read and write?
2. What are the theories that are shaping coding literacy?
3. Does the current data on learning to code and learning to read and write intersect and what theories emerge from the two?

Theoretical Framework

Computer coding, robotics, and educational technology is based on Papert's theory of constructionism (1980), includes discovery learning, collaborative learning, problem solving, project-based learning, competition based learning, and compulsory learning. These approaches follow the ideas of constructionism introduced by Papert (1980) and constructivism derived from Piaget's work (as cited in Papert, 1980). In his Theory of Cognitive Learning, Piaget argued that people produce knowledge and form meaning based upon their experiences. This theory covered learning theories, teaching methods, and education reform. Two of the key components that create the construction of an individual's new knowledge are accommodation and assimilation. Constructionism as an educational theory is student-centered and emphasizes discovery learning, during which students are encouraged to work with tangible objects in the real world and use what they already know to gain more knowledge. The point is to make the process of thinking and learning visible and to allow for a more process-oriented engagement with an idea via construction and deconstruction. Under constructionism, teachers take a backseat role as facilitators of student learning instead of giving lectures or step-by-step instruction (Altin & Pedaste, 2013; Foreman & Kuschner, 1983; Papert, 1980). I studied the role of learning to code and how that skill can impact literacy skills in kindergarten through twelfth grade students, including learning to read and write through the lens of Seymour Papert's Constructionism Theory.

Data Analysis Procedures: A Structure for Qualitative Content Analysis to Uncover Emerging Grounded Theory

Qualitative Content Analysis procedures. According to Neuendorf (2017), the design of a Qualitative Content Analysis follows the standards of the scientific method. Neuendorf (2017) suggested clearly defined steps in the Qualitative Content Analysis process. The following steps were implemented in this study:

1. Objectivity as well as intersubjectivity was established. I conducted a Qualitative Content Analysis through a secondary Grounded Theory lens, the question that I continuously asked during this study was: What theory or theories do I see developing from the literature examined on coding and literacy? The thoughts and assumptions regarding this subject were put aside.
2. A search of databases was conducted to include the topics of coding, literacy, reading, writing, computer science, developmental stages of reading and writing, the developmental stages of learning to code, and all variations of these topics. Once I found saturation, the data was examined.
3. I reviewed articles to discover significant statements.
4. Coding and themes were discovered in alignment with the three sources while being reviewed.
5. A determination of theories that emerged from the data were cross examined for reliability and validity.
6. Through the discovery of the theories that evolved from examining coding and the connection to the developmental stages of learning to read according to Chall (1976), and the developmental stages of learning to write according to Graves (1994), I established generalizability and replicability.

My first step in this Qualitative Content Analysis was to formulate research questions. Those questions were replaced by open-ended questions that guided my research and influenced the data that was gathered. In this qualitative study, I read, analyzed, and interpreted the data for concepts and patterns. As a result of this process, some patterns and concepts emerged that were not previously indicated, nevertheless, they were important concepts to consider in the theory development (Neuendorf, 2017).

Theoretical sampling was the next step in the Qualitative Content Analysis process. This particular sampling process consisted of data collection for generating theory whereby I jointly collected, coded, and analyzed my data and decided what data to collect next and where to find them, in order to develop the theory as it emerged. The process of data collection was guided and controlled by the emerging theory (Glaser & Strauss, 1965).

When implementing Qualitative Content Analysis as a research method, I systematically and objectively analyzed a sampling of the written, verbal, and visual communications produced within the last 10 years on literacy, computer coding, and robotics instruction in the kindergarten through twelfth grade school setting.

Data analysis procedures using Grounded Theory as a lens. A Grounded Theory study goes beyond conjecture and preconception to the underlying processes of the phenomena being studied. The rationale behind selecting Grounded Theory as a secondary lens for this study was to delimit the study to the primary concern and focus on the influence that developmental reading and writing skills have on coding literacy skills. As a secondary lens, Grounded Theory allowed the data to emerge from the three selected data sources. Also, delimitation will prevent preconceived ideas to mask what actually occurred in the field of coding and traditional literacy, and instead allowed patterns to emerge from the data. Grounded Theory methods consist of systematic guidelines that ebb and flow throughout the collection and analysis of qualitative data. As the data were collected and analyzed, I then sorted the themes from the data to develop a theory. This theory emerged from the comparative methods used in the examination of the data collected. Grounded Theory leads researchers to attend to what they hear, see, and sense while gathering data (Urquhart, 2013).

According to Urquhart (2013), the defining components of Grounded Theory included and the steps that I followed were:

1. Collected and analyzed data simultaneously to develop a theory.
2. Constructed analytic codes and themes from data, not from personal, preconceived ideas and thoughts.
3. Reviewed scholarly journals for significant statements.
4. Implemented the constant comparative method by making comparisons during each stage of analysis.
5. Used the constant advancement of theory development during each step of data collection and analysis.
6. During memo-writing, I stopped and analyzed ideas that were developing and emerging in the data through my method of coding.
7. Applied theoretical sampling that was aimed towards theory construction.
8. Conducted the literature review after the independent analysis.

When exploring emerging theories, I focused on how individuals interacted with the phenomena being studied. In the current study, the phenomenon was the correlation of the developmental stages of learning to read and write and the developmental stages of coding literacy. Grounded Theory as a secondary lens, allowed me to ground this study in the data in a secondary method that any theory produced is verifiable. The concept of Grounded Theory suggests that the researcher should seek to understand what the data indicates, rather than trying to manipulate the data to fit into a particular existing theory. Data collection and interpretation varied throughout the research study (Patton, 2002; Urquhart, 2013).

The use of Grounded Theory as a secondary lens is flexible because I was able to vary the interpretation of data based on emerging or established analyses as situations changed, or as further data were analyzed. By applying Glaser's (1992) comparative method of Grounded Theory as a secondary lens, I was able to implement various strategies for data analysis that were efficient, and productive without forcing the data into a pre-determined formula. Glaser and Strauss (1965) state that, "Theory must fit the situation being researched, and work when put into use" (p. 21).

Grounded Theory has integrity as a research process because it does not seek to impose preconceived thoughts and ideas on the acquired data. The concept of overlapping data collection and analysis were used for this study. This inductive process required constant review of the data in order to determine the next sampling. This was an example of theoretical sampling because the emerging data and themes determined future data collection (Urquhart, 2013).

Description of Research Sites

The research sites for this study, since it is a Qualitative Content Analysis, consisted of multiple sources. I collected scholarly articles through the university library online resource system. Standards I examined included the K-12 United States Computer Science Framework (2016), Computer Science For All (2018), International Standards for Computer Science Educators (2016), Common Core State Standards (2009), and Computer Science Teachers Association Standards (2017). In addition, the following coding websites were thoroughly researched: CSunplugged (Computer Science unplugged), Code.org/ Hour of Code, 2 simple to code, and Code Academy.

Role of Researcher

Qualitative research is often overwhelming to a researcher, particularly to new researchers, who must learn to organize, and interpret all of the data sources. Within this Qualitative Content Analysis study, I maintained an open mind and allowed patterns of the data to naturally emerge. In that sense, I was assumption free. As I searched the topics of coding and how it related to learning to read and write, I did not know specifically a priori what theory would emerge from the data sources. The themes and the core variables emerged as the study progressed (Urquhart, 2013). My role was to complete a thorough Qualitative Content Analysis and emerging Grounded Theory for the relationship between learning to code and developing reading and writing skills. Through the process of open coding, themes and structures within the data collection pieces were sorted and analyzed. The purpose of using memoing as a cornerstone in this research was to exemplify quality and to determine when saturation of the sources had been identified. According to Stern (2007), memos are the mortar that hold together the building blocks(data) that comprise a Grounded Theory study. In this particular study, the researcher memoed notes from articles that were focused on the topic of literacy, coding, and computer science. In addition, I read and scripted memos from articles and books that contained Qualitative Content Analyses and Grounded Theory studies. Theoretical sampling guided how the data was generated, and from the sources and locations, as the study moved forward and progressed.

Data Collection Procedures

While gathering data for this study, I read professional journal articles, examined various websites that promoted learning to code with kindergarten through twelfth grade students, and studied coding standards for the inclusion of coding skills. Databases were searched thoroughly for scholarly articles related to the topics of coding as a literacy, reimaging the role of technology in education, the relationship between coding and writing, effects of technology on literacy skills, computational thinking, and standards which included the topics of coding, literacy, and computer science. The articles and books were organized, recorded, and filtered in a research matrix for coding as it related to the following questions that I answered in this study:

1. What theory emerged that connected learning to code and coding literacy to the developmental stages of learning to read and write?
2. What are the theories that are shaping coding literacy?
3. Do the standards on learning to code and learning to read and write intersect and what theories emerge from the two?

Data source #1- scholarly articles and books. While examining data sources, an extensive review of scholarly articles and books related to the topic of coding and literacy were discovered and reviewed. The criteria for selecting the articles included:

1. The article, or book must have been written between the years of 1990 and the current year.
2. The article or book must have explored a link or relationship between reading and writing literacy and computer coding literacy.

3. The article or book must have been peer reviewed and contain the topics of literacy and computer coding.

Through reading the articles that meet the selected criteria, I explored the emerging theories that developed. In addition, I examined the articles for themes and ideas that lead to further connections and research exploration. The questions that I answered were:

1. What theory emerged that connected learning to code and coding literacy to the developmental stages of learning to read and write?
2. What theories are shaping coding literacy in scholarly journals and books?
3. Does the current data on learning to code and learning to read and write intersect and what theories emerge from the two throughout the articles and books?

While examining scholarly articles, I also determined the stages of development in reading and writing, and if those stages correlated with the progressive stages of learning to code on the computer. In this study, I established a set of terms and examined each data source to determine if the identified terms existed in each source and the percentage of sources that included the identified terms. Through this examination and determination of frequency, I discovered the theories that emerged from the data sources. The stages of reading development according to Chall (1976), and the stages of writing development according to Graves (1994) were compared to the stages of the development of learning to code. The first question in this research study guided this inquiry: What theory emerged that connected learning to code and coding literacy to the developmental stages of learning to read and write? Scholarly journals included were, but were not limited to: *The International Journal of Learning*, *Educational Researcher*, *Technology, Knowledge and Learning*, *Computers & Education*, *Journal of Research on Technology in Education*, *The Reading Teacher*, *Educational Technology Research and Development*, *Journal of Educational Computing Research*, *Journal of Literacy Research*, *Reading Research Quarterly*, *International Journal of Technology and Design Education*, *Journal of Information Technology Education: Innovations in Practice and Computers and Education.* During my review of all of these scholarly journals, I uncovered the fact that the multitude of articles that incorporated my research topic were not included in traditional literacy journals. Traditional literacy journals had articles related to reading and writing development and incorporating technology into literacy instruction. My focused research topic was about aligning coding/programming on a computer to the steps required in learning to read and write. In order to answer my research questions, I needed to find the alignment of the two.

Data source #2-technology/coding curriculums and standards. The second data source consisted of local, state and, national technology/coding standards related to computer science, programming, and robotics. These were examined to determine the extent to which coding is currently taught in elementary schools, the future plans for incorporating coding into the curriculum, and the connection of teaching coding to reading and writing skill development. The sub question #2 in this research study guided this inquiry: Does the current data on learning to code and learning to read and write intersect, and what theories emerged from the two?

Table 2

States with K-12 Standards for CS Content

State	Year Adopted	Grade Bands or Individual Grades	Location
Arkansas	2016	K-8; individual grades 9-12; grade bands	CS
Florida	2016	K-2, 3-5, 6-8, 9-12	Science, CTE
Idaho	2017	K-2, 3-5, 6-8, 9-10, 11-12	CS, CTE
Indiana	2016	K-2, 3-5, 6-8, 9-12 (CTE only)	Science, CTE
Massachusetts	2016	K-2, 2-5, 6-8, 9-12	Digital Literacy and CS, CTE
New Jersey	2014	K-2, 3-5, 6-8, 9-12	Technology, CTE
Washington	2016	K-2, 3-5, 6-8, 9-12	CS, CTE

Note. Adapted from *State of the states' landscape report: State-level policies supporting equitable K-12 computer science education* (2017). CS stands for Computer Science. CTE is xx.

The table above indicates the states within the United States that currently have Computer Science standards included in their state required curriculum. In addition, the standards from the K-12 United States Computer Science Framework (2016), Computer Science For All (2018), International Standards for Computer Science Educators (2016), Common Core (2009), and Computer Science Teachers Association Standards (2017) were examined.

I identified a set of terms and examined each data source to determine if the identified terms existed in each source and the percentage that the terms existed. Based upon this examination and the identification of frequency, I discovered the theories that emerged from the data sources. By examining coding standards, I also determined the stages of development in reading and writing, and if those stages correlated with the progressive stages of learning to code on the computer.

Data source #3- websites that encourage or teach computer coding to kindergarten through twelfth grade students. The third data source consisted of websites that promoted learning to code in elementary schools and why the skill was considered to be imperative to a student's success. The criteria for including a website into the data source included:

1. The website has an instructional component that teaches individuals how to code on the computer.
2. The website is current and active and includes a section that applies specifically to kindergarten through twelfth grade students.
3. The website is endorsed by a reputable computer or educational company, or organization to provide authentic coding instruction.

There are several such websites, including one developed by the CEO of Google, one created by Apple Inc., and Code.org. Specific online resources that were reviewed included: (a) United States Department of Education Computer Science, (b) Code.org, (c) Hour of Code, (d) Code Academy, and (e) Computer Science for All.

The sub question #3 in this research study guided this inquiry: Do the websites on learning to code and learning to read and write intersect and what theories emerged from the two? While examining learning to code websites, I also analyzed the stages of development in reading and writing and if those stages correlated with the progressive stages of learning to code on the computer. In this study, I identified a set of terms and examined each data source and determined if the identified terms existed in each source and what percentage it existed. Through this examination and the determination of frequency, I discovered the theories that emerged from the data sources.

Data Recording/Analyzing Procedures

Each data source was processed through a series of steps to ensure validity and reliability. The data were organized into categories of information, and themes across the sources were established. I applied a coding process, which according to Saldaña, Saldaña, and Miles (2015), “In a qualitative inquiry, to apply a coding process is to identify a word or short phrase that symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute for a portion of language-based or visual data” (p. 3). Since I took an inductive approach to my study, the coding process also developed as the research progressed. The coding system was more grounded empirically, therefore connected the research method of Grounded Theory through a Qualitative Content Analysis. The data sources examined were scholarly articles on the topics of coding literacy and the developmental stages of reading and writing, a sample of coding standards from state and national levels, and websites that taught coding literacy.

In desegregating the data, I first determined the significant statements and specific clusters of meaning related to literacy and coding. With the unique terms related to coding and computer programming, I implemented an in vivo coding structure. The root meaning of in vivo is, in that which is alive, and as a code refers to a word or short phrase from the actual language found in the qualitative data record, and the terms used by participants themselves. In vivo coding is also one of the methods recommended to apply to Grounded Theory research. Charmaz (2006) identified three types of in vivo code: general terms that flag some significant meaning, codes that can be participants’ innovative terms, or insider shorthand terms as in those used in organizational settings (Charmaz, 2006; Strauss, 1987).

Once I reached saturation and determined that there were not any new concepts emerging from the data, the analysis process began. In the second cycle, I took the findings of all three data sources combined and determined the final results of the study and identified the theories that emerged.

Data Presentation

The essence of Grounded Theory is the constant comparative method in conjunction with theoretical sampling. The constant comparative method was implemented to review and present the data gathered in this study. Constant comparative coding combines the approaches of an explicit coding procedure in alignment with the development of a theory. In Grounded Theory development, data collection, coding and analysis are completed simultaneously. The purpose of this alignment and analysis is to generate theory more systematically. Constant comparative coding method was also used jointly with theoretical sampling. When using Grounded Theory, it is necessary to acquire overlapping data collection and analysis. Constant comparative coding means that the emerging concepts from the current data were examined to decide where to sample from next (Charmaz, 2014; Glaser & Strauss, 1967).

Charmaz (2006) suggests that theoretical sampling allows for two things to occur: it enables a researcher to build up justification for concepts in the theory by finding more examples of a particular concept, and it also allows researchers to follow an emerging storyline suggested by the data. The process of constant comparison analysis is to constantly compare instances of data labeled as a particular category with other instances of data in the same category (Urquhart, 2013).

Trustworthiness

In order to establish trustworthiness, I put aside all pre-determined notions, ideas, or opinions while I evaluated the data. I took into consideration the pre-understanding during the planning process, as well as the analyzing process in order to minimize any biases. As a final check, I considered how the new findings corresponded to the literature and whether or not the result was reasonable and logical (Burnard, 1991; Morse & Richards, 2002; Urquhart, 2013). Utilizing in vivo coding strengthened the authenticity of the results. The in vivo codes are significant when interpreting the data directly from participants, journals, websites, and coding standards (Saldaña, Saldaña, and Miles, 2015).

Ethical Issues

To conduct a thorough study is to conduct an ethical study. Critical self-reflection on biases and theoretical dispositions existed throughout the study. It was my responsibility to report, and analyze the data honestly, and to take responsibility for reporting the finding accurately. Future research and implications could be impacted by the findings that are reported. In order for the study to be considered reliable and valid, all current research must be (and was) examined and analyzed to produce generalized results (Urquhart, 2013).

Chapter Summary

In summary, I researched if computer coding was a necessary skill that kindergarten through twelfth grade students should be taught, and if the skill of coding contributed to the development of reading and writing skills. This study was an example of qualitative research. The qualitative methodological framework utilized for this study was Qualitative Content Analysis with a secondary lens of Grounded Theory. There were three questions to be answered through this study:

1. What theory emerged that connected learning to code and coding literacy to the developmental stages of learning to read and write?
2. What are the theories that are shaping coding literacy?
3. Does the current data on learning to code and learning to read and write intersect and what theories emerged from the two?

In the next chapter, I analyzed the data collected to uncover the core message of the data sources.

CHAPTER IV

Analysis of Data

The primary purpose of this study was to determine what theories might emerge around the topics of learning to code and learning to read and write. The intent of this research project was to illuminate the literacy of computer coding. Educators and programmers have previously made the connection between programming and literacy (Clements, 1999; Hutchinson, 2015; Kazaoff & Bers, 2014; Kazakoff, Sullivan, & Bers, 2012; Nadolny & Estapa, 2015, Papert, 1980; Rich, Browning, Perkins, Shoop, Yoshikawa, & Belikov, 2018; Snow, Tabors, Nicholson, & Kurland, 1994). Many previous studies on how learning to code can improve science, computer science, and STEM related skills and interests in students exists (Barker & Ansorge, 2007; Bers, 2018; Chitterji, 2018; Clements, 1999; diSessa, 2001; Fayer, Lacey, & Watson, 2017; Geist, 2016; Kafai & Burke, 2016; Khine, 2017; Papert, 1993, 1987; Phillips & Brooks, 2017; Sergeyev, Alaraje, Meyer, Kinney, & Highum, 2015; Sullivan & Bers, 2015). However, most published research does not connect coding literacy to reading and writing literacy.

Computer code is layered within the technology of writing and structures much of our modern-day communications, including word processing, e-mail, the Internet, social network platforms, digital video production, and mobile phone technology (Fayer et al.,2017; Rich et al., 2018; Vee, 2017). The terms coding and programming were used interchangeably throughout this study. Programming and writing have a complicated association. Programming is considered a form of writing because, symbols are used to create a message to read, although, programming is more than writing. It is a planned sequence of symbols that will be read, executed, or run, by the computer (Vee, 2017).

The purpose of studying the literature and its findings, the computer science standards, and coding websites, was to determine if learning to code can have an impact on literacy skills, specifically developmental reading and writing skills. This chapter presents the data and findings from the three data sources; (a) four scholarly journals and articles, (b) five sets of standards that incorporate the teaching of coding/programming, and (c) four websites that teach coding. This chapter contains the findings from each journal, each set of standards, and each website. After reviewing the data from each source, I summarized the composite findings from that source. To summarize all three data sources, I combined and filtered the overarching themes from all three data sources combined to determine the theories that emerged. These findings are explained in Chapter 5. This chapter explains the similarities between learning how to code/program on a computer and learning to read and write. This chapter also reveals specific data found in each data source, and how the data sources combine.

Data Analysis of Data Source 1: Scholarly Journals

Cycle 1: Sample selection journals

I surveyed nine scholarly journals that incorporated coding and computer science into K-12th grade curriculums. For the years 2007-2017, I collected the table of contents for every issue included in each journal of the nine identified: (a) *Computer Science Education*, (b) *International Journal of Computer Science Education in Schools*, (c) *Technology Knowledge and Learning, Education Technology Research and Development*, (d) *Themes in Science and Technology Education*, (e) *Computers and*

Education, (f) *Journal of Education Computing Research*, (g) *TechTrends*, and (h) *Journal of Research on Technology and Education*. I reviewed the table of contents for entries that included: (a) literacy, (b) reading, (c) writing, (d) coding/programming, and (e) robotics. Out of 5,242 journal articles, only 130 of those articles from the years 2007-2017 referenced literacy as it relates to reading and writing. The results of the initial journal search for articles are as follows: *Computers and Education*, N= 2,356; *TechTrends*, N=1,143; *Educational Technology Research and Development*, N=598; *Journal of Educational Computing Research*, N=452; *Journal on Research and Technology Education*, N=209; *Technology, Knowledge and Learning*, N=195; *Computer Science Education*, N=171; and *Themes in Science Technology Education* N=93; *International Journal of Computer Science Education in Schools,* N=25. During my review of all of these scholarly journals, I uncovered the fact that the multitude of articles that incorporated my research topic were not included in traditional literacy journals. Traditional literacy journals contained articles related to reading and writing development and incorporating technology into literacy instruction. My focused research topic and questions revolved around aligning coding/programming on a computer to the steps required in learning to read and write. In order to answer my research questions, I needed to find the alignment of the two.

Cycle 2: Journal article selection

After analyzing the journals, I selected four journals from Cycle 1 that incorporated the most articles that related to my research topic of coding and literacy. My intentions behind selecting the four journals was to examine the journals that contained the most articles related to the specific terms that I identified for my research.

Out of the nine journals examined, it was determined that four of the journals included the greatest number of articles with the identified key words: these four journals included: (a) *Computers and Education*, N=95; (b) *Tech Trends*, N=23; (c) *Journal of Education Computing Research*, N=20; (d) *Journal of Research and Technology in Education*, N=8. Within these four journals, I found the most articles related to my topic. For the purpose of this research, I only wanted to include that focused on Kindergarten through 12th grade students. By selecting these four journals, I completed the second cycle of coding. The specific number of articles with each code word identified are labeled in Table 3, Appendix A. Specifically, refer to the first four journals identified in Table 3 in Appendix A.

After identifying the four journals, I created a matrix and catalogued all the articles included between the years of 2007-2017 and coded using the terms: (a) literacy, (b) reading, (c) writing, (d) coding/programming, and (e) robotics. The following journals did not have enough relevant articles to include in the remainder of the coding cycles and were not included in the analysis: (a) *Computer Science Education*, N=18; (b) *Journal of Research and Technology in Education*, N=8; (c) *International Journal of Computer Science Education in Schools*, N=8; (d) *Themes in Science and Technology Education*, N=7; and (e) *Technology Knowledge and Learning,* N=6. It was within the other four top journals that I found a significant number of articles to review. The journal titled *Computers and Education*, included N=95. Within the journal *Educational Technology Research and Development*, N=19. For the *Journal of Educational Computing Research* N=20, and *Tec Trends*, N=23. In total for my Qualitative Content Analysis, I identified 157 scholarly articles. By completing this process and identifying the scholarly journals and articles for Data Source 1, I narrowed down my sample selection. I determined my sample selection of articles by cataloging the number of code words that were included in each of the 157 articles. The code words identified are reading, writing, literacy, programming/coding, and robotics. I then narrowed down those 157 articles to 32 that related the closest to my topic of coding and literacy. As I narrowed the selection of articles, I looked specifically for articles with a direct connection to computer programming and the traditional definition of literacy, which is the ability to read and write.

Cycle 3: Individual Journal Analysis

Analysis procedures of scholarly journals. Now with my selected articles identified it was time for Cycle 3. During the third cycle of coding, I read each of the 32 articles in detail. There were additionally three separate cycles that I worked through on each article. First, I read the article for significant statements. For the purpose of this study the significant statements refer to the identification of statements within the data source that signify important points related to the topic of coding/programming and literacy. Secondly, I identified 'clusters of meaning' within the body of the article that related to my topic of research. For the purpose of this study clusters of meaning refer to the identification of a group of words that are related to the research topic Thirdly, I collected, sorted, and discovered the emerging theories from the clusters of meaning statements that answered my overarching question.

Journal 1: *Computer & Education*. A continued push for computer science education is the reoccurring theme throughout the articles in *Computers & Education*. In Table 4 of Appendix B, 16 articles were coded for significant words, clusters and themes that emerged from each one. Robotics programming, computational thinking, and the implementation of technology in the classroom purport more data that supports the connection of literacy skills and the programming/coding. The Transformative Robotics Experience for Elementary Students project is one of the many initiatives bringing robotics and computer science curriculum to elementary age students (Chen, Shen, Barth-Cohen, Jiang, Huang, & Eltoukhy, 2017). The project used the humanoid robot platform NAO by Aldebaran Robotics. It contains tools such as voice/speech and facial recognition which is similar to robotics platforms used by current professionals. In the journal of *Computers & Education* there were 15 themes that emerged throughout the articles examined. The overarching clusters of meaning were selected by reading each article and highlighting the phrases that connected to literacy, specifically reading, writing and digital literacy. In total 100 clusters of meaning were identified within articles pulled from 2007-2017 that related to my topic. In Table 4 of Appendix B referring specifically to the *Computers & Education* the clusters of meaning are explicitly listed for each article. The frequency of the clusters of meaning discovered included: (a) reading comprehension, N= 13; (b) problem solving, N= 11; (c) vocabulary, N= 10; (d) listening comprehension, N= 8; (e) literacy N=8; (f) reading, N=8; (g) communication skills, N= 7; (h) sequential analysis, N= 7; (I) coding/programming, N=6; (j) cross curricular connections and intersections, N=6; (k) robotics, N=6; (l) writing, N=5; (m), computational thinking, N=3; and (n) engineering, N=2. In the last step, I synthesized the original clusters of meaning groups to create the following overall clusters themes: literacy and reading comprehension, N=21; writing, sequential analysis, and communication skills, N=19; problem solving and computational thinking, N=14; robotics and engineering, N=8; and coding/programming, N=6.

Cluster Themes for Journal 1

Literacy and reading comprehension. Investigating the progression of reading comprehension skills (N=21) was the most common cluster of meaning extracted from the volumes of *Computers & Education* journal. In a contributing study by Lysenko and Abrami, the impact of two web-based applications on elementary students' reading comprehension was measured (2014). This research consisted of two studies conducted during 2010-2011 and 2011-2012 school years, participants included 26 teachers from grades 1-2, and their students (N=517). Analyses showed that students who used both web-based application tools performed significantly better ($p < .001$) compared to controls in reading and writing as measured by standardized tests (Lysenko & Abrami, 2014).

Writing, sequential analysis, and communication skills. The second largest theme emerged from the cluster groups of writing, sequential analysis, and communication skills (N=19). An example of this theme can be seen in the study conducted in 2013 by Genlott and Gronlund is an example of the writing, sequencing and communication skills themes. The authors analyzed the effects of ICT (Information and communication technologies) supported "Integrated Write to Learn" (iWTR) method in which first grade students used computers and other ICT tools to write texts and discuss and revise them with classmates and teachers. The method was tested using two testing groups and two control groups. The study measured the performance in reading and writing using standard tests in combination with observations and student evaluations.

The results showed that reading skills improved, but the biggest improvement was with writing skills. The students in the test groups wrote texts with better grammar, and sentence structure, more defined content, and a larger vocabulary (Genlott & Gronlund, 2013).

Problem solving and computational thinking. Examples of studies that incorporated the themes of problem solving and computational thinking (N=14) from the *Computers & Education* are shared in this paragraph. In an article written by Chen, Shen, Barth-Cohen, Jiang, Huang, and Eltoukhy an instrument was created and administered that measured the computational thinking of fifth grade students as they learned coding in robotics and everyday reasoning abilities (2017). The authors determined that participating in a robotics curriculum did students improve their computational thinking skills. In another study conducted by Potocki, Ecalle, and Magnan, (2013), two groups of second graders in a randomized control trial design were selected and trained using two computer assisted comprehension training programs (CAI). Two different CAI programs were compared. Through the use of these programs students listening and reading comprehension skills improved (Potocki, Ecalle & Magnan, 2013).

Robotics and engineering. The term robotics yielded a frequent cluster of meaning (N=8) because in order to participate in a robotics curriculum a student must be able to sequence, problem solve, comprehend through reading and writing, and program/code a computer to activate the movement of the robot. Within the articles in *Computers & Education* authors frequently, sometimes outwardly, spoke to the similarities and other times inferred the connection between the developmental steps of learning to read and write and the steps needed to code a robot. Kucuk and Sisman, contributed an article in 2017, where the purpose was to understand the behavioral patterns of elementary students and teachers who participated in a one-to-one robotics curriculum and, identified the connection to learning sequencing skills.

***Coding/programmin*g**. The clusters of meaning with the fewest frequencies found in *Computers & Education* included computer programming/coding (N=6) which is considered an important skill for the development of higher order thinking (Fessakis, Gouli, & Mavroudi, 2012). An example from the journal presented, is an exploratory case study featuring 5-6-year-old kindergarten children exposed to a short introductory programming game. Then the children had to solve a series of computer programming problems. The results of the study showed that when young children work with a real programming environment, they learn how to execute commands and respond positively to the immediate feedback (Fessakis, Gouli, & Mavroudi, 2012).

Core themes for the journal Computers & Education. In total 100 clusters of meaning were identified within the articles pulled from 2007-2017 related to my topic. In the journal of *Computers & Education* there were 14 themes that emerged throughout the articles examined. As I reviewed the 10-year span of the table of contents, I identified articles to review. I read each selected article for clusters of meaning, and began to see patterns forming and clusters of themes developing. I was looking to determine if there was an overall connection between computer programming/coding and the skills needed to learn how to read and write. The themes that emerged consolidated into two overarching themes. The first overarching theme was preliminary skills necessary in order to acquire the ability to comprehend what a student is reading (vocabulary, literacy, reading, sequential analysis and writing). The second overarching theme that evolved was digital literacy (coding, programming, robotics, computational thinking, communication skills, and problem solving.) *Computers & Education* seeks to promote not only the necessity of incorporating technology into the classroom, but also the research behind how becoming digitally literate can benefit students in other subject areas.

Journal 2: *Educational Technology Research and Development*. The articles included in Educational Technology Research and Development focus on the topics of reading skills and applying reading skills to learning via the computer. In Table 5 of Appendix C, I noted that five articles underwent coding for significant words, clusters and themes. In total 146 clusters of meaning were highlighted. The overarching clusters of meaning were selected by reading each article and highlighting the phrases that connected to literacy, specifically reading, writing and digital writing. Many of the articles centered around incorporating and integrated technology into the general everyday curriculum. This journal offered a multitude of strategies based on peer reviewed studies that incorporated programming and utilizing computer programs to supplement instruction from a classroom teacher.

Within *Educational Technology Research and Development,* a total of 39 clusters of meaning were identified. Those clusters included; (a) writing, N=7; (b) reading skills, N=5; (c) sequencing, N=5; (d) literacy, N=4; (e) reading comprehension, N=4; (f) decoding, N=3; (g) vocabulary, N=3; (h) collaborative learning, N=2; (I) language arts/cross curriculum, N=2; (j) programming, N=2; (k) problem solving, N=1; and (l) self-efficacy, N=1. In the last step I synthesized the individual clusters groups into the following themes: reading, writing and literacy, N=16; reading comprehension, vocabulary, decoding, and sequencing, N=15; programming, and problem solving, collaborative learning and self-efficacy, N=6; and language arts/cross curriculum, N=2.

Cluster Themes for Journal 2

Reading, writing and literacy. Investigating the progression of reading, writing and literacy (N=16) was the most common cluster of meaning extracted from the volumes of *Educational Technology Research and Development*. In a study included in *Educational Technology Research and Development,* the authors investigated the effects of multimedia story reading and questioning on children's literacy skills, including vocabulary, story comprehension and reading engagement, Zhou and Yadav, (2017). The results showed significant interaction of media and questioning on targeting vocabulary and effects for engagement. Literacy skills, such as decoding, sequencing, and predicting have been identified as key predictors in children's future academic success (Fletcher and Reese, 2005). Researchers and teachers are always in search of new strategies to effectively teach reading skills. Recently, sophisticated multimedia devices have provided new ways to introduce children to reading and early literacy skills.

Reading comprehension, vocabulary, decoding and sequencing. The original cluster groups of reading comprehension, vocabulary, decoding and sequencing (N=15) were combined to create the second largest theme. An example of this theme can be found in Ponce, Mayer, & Lopez's (2013) study that explored the effectiveness of a computer based spatial learning strategy for improving reading comprehension and writing skills. In the area of reading comprehension, students received practice in transferring information from passages read into graphic organizers. In addition, students also practiced the planning stages for writing by filling in graphic organizers and translating them into paragraphs. The findings indicated that students who participated improved their reading and writing skills significantly more than the study group that did not apply the computer based spatial learning strategy.

Programming, problem solving, collaborative learning and self-efficacy. The original cluster groups of programming, problem solving, collaborative learning and self-efficacy (N=6) formed the next theme. Computer programming requires problem solving strategies and logical thinking (Wang and Hwang, 2017). Research supports the belief that collaborative learning is an effective strategy for teaching, by encouraging student interaction and offering a variance to the explanation of content. Although, students must be taught and monitored to ensure the productivity of collaborative learning, programming can provide opportunities for students to problem solve in groups and learn from each other (Wang and Hwang, 2017). In addition, research showed that students who participated in collaborative learning activities, had higher self-efficacy and a lower cognitive load than students working individually on the same activity (Wang and Hwang, 2017).

Language arts/cross curriculum. The smallest theme that emerged from the cluster groups was language arts/cross curriculum (N=2). In a 2010 study conducted by Judson; he investigated the question of does improving technology literacy open doors to traditional content across subject areas. This particular study focused on technology literacy and how it relates to language arts skill development and overall student achievement. The study included approximately 5,000 fourth and fifth grade students and 5,000 sixth to eighth grade students were studied. Judson's study, featured in *Educational Technology Research and Development,* directly looked at whether or not there is an identifiable link between gains in technology literacy and achievement in the academic areas of reading, mathematics and language arts. The changes were compared to relative gains from a pre- to post-assessment in technology literacy. Specifically, this study was grounded in two ideas: that technology gains lead to more confidence in learning the content subject matter, and that technology gains reflect the increased ability to use technology as a mediator of new learning. The results provided evidence that connections between technology literacy gains do correlate with gains in language arts skills (Judson, 2010).

Core themes for the journal of Educational Technology Research and Development. In total, I identified 39 clusters of meaning within the articles pulled from 2007-2017 related to my topic. In *Educational Technology Research and Development,* there were 12 themes that emerged from the examined articles. As I reviewed the ten-year span of the table of contents, I identified articles to review, and read each selected article for clusters of meaning. I was reading to determine if there was an overall connection between computer programming/coding and the skills necessary to learn how to read and write. The themes that emerged consolidated into two overarching themes. The first overarching theme was cross-curriculum integration. Many of the articles discussed how coding/programming could be applied to a range of subject areas including language arts. What I found especially interesting about the direction this journal took was that many of the studies were based in the classroom and applied communication skills as well as group problem solving strategies, and cooperative team-based learning. The second overarching theme from *Educational Technology Research and Development,* was the overall idea of technology literacy and how it can be applied to the development of a well-rounded learning program. This journal had many articles that specifically referenced scaffolded practice and cognitive development.

Journal 3: *Journal of Educational Computing Research*. Inside the *Journal of Educational Computing Research*, I found articles on digital writing, computer guided oral reading, pedagogical programs for learning, computational thinking, and early cognitive development all in relation to computer coding/programming. In total, I coded four articles, (see Table 6 in Appendix D). The themes of sequencing skills, language and vocabulary development, and visual memory skills continued to weave throughout the research. In 2014, there was a study published in this journal on the impact of programming robots to teach sequencing skills to thirty-four early childhood students (Kazakoff, and Bers, 2014). The students' sequencing skills were assessed before and after the intervention and the scores were comparted using a t-test. A significant increase in post-test scores comparted to pre-test scores was found (Kazakoff, & Bers, 2014).

The clusters of meaning found in the *Journal of Educational Computing Research* included a more varied list of topics. The clusters of meaning included; (a)writing, N=4; (b) decoding, N=2; (c) literacy, N=2; (d) programming, N=2; (e) reading, N=2; (f) reading comprehension, N=2; (g) sequencing, N=2; (h) e-books, N=1; (I) metacognitive thinking, N=1; (j) phonics, N=1; (k) problem solving, N=1; (l) visual discrimination, N=1; and (m) vocabulary, N=1. The clusters of meaning are specifically listed in Table 5 of the appendix. The total number was 22. In the last step I synthesized the individual clusters groups into the following themes: writing, sequencing and literacy, N=8; reading, reading comprehension, E-books and vocabulary, N=6; programming, metacognitive thinking and problem solving, N=4; and decoding, phonics and visual discrimination, N=4.

Cluster Themes for Journal 3

Writing, sequencing and literacy. Investigating the progression of writing, sequencing and literacy (N=8) was the most common cluster of meaning. An example of this theme focused on the connection between writing and technology through an examination of whether an automated essay feedback system could improve the writing performance of college students (Kellogg, Whiteford, & Quinlan, 2010). A struggle often noted by college professors is the weak writing skills of new college students. In this study, students in a freshman composition course were given no feedback on the first drafts of three practice essays. This study examined the effects of no feedback, intermittent feedback, or continuous feedback. Transfer tests showed that students learned to reduce errors of mechanics, usage, grammar and style. The benefit was determined to be for continuous feedback, but not from intermittent feedback (Kellogg, Whiteford, & Quinlan, 2010).

Reading, reading comprehension, E-books and vocabulary. The original cluster groups of reading, reading comprehension, E-books and vocabulary (N=6) were combined to create the second largest theme. In 2012, an article by Korat and Shamir, in the *Journal of Educational Computing Research,* examined the effects of direct and indirect teaching of vocabulary and word reading on pre-kindergarten and kindergarten children following the use of an electronic storybook (E-book). Children in each age group were assigned randomly to either the control group given the regular school program, or the intervention group which read the e-book. The e-book included vocabulary support. The children who read the e-book showed progress in the comprehension and the reading of words directly supported by the e-book compared to the control group (Korat & Shamir, 2012). The level of vocabulary development can be a key factor in overall communication skills (Vermeer, 2001), particularly in reading and reading comprehension (Nagy & Scott, 2000; Stanovich, 1986).

Programming, metacognitive thinking, and problem solving. One of the smallest themes that emerged from the cluster groups was programming, metacognitive thinking, and problem solving (N=4). An example of a study that assessed the impact of meta-cognitive training on students' understanding of basic programming concepts was published in the *Journal of Educational Computing Research* in 2014 by the authors, Cetin, Sendurur, & Sendurur. Their study investigated meta-cognitive training on students' programming achievement, retention of programming knowledge, and explore students' experiences and opinions related to meta-cognitive training (Cetin, Sendurur, E. & Sendurur, P., 2014). The 51 participants were divided between an experimental group that received instruction in a meta-cognitive model and a control group taught using a traditional method. The results indicated that the instruction based on meta-cognition produced significantly better acquisition of introductory programming concepts.

Decoding, phonics and visual discrimination. The final, theme with the least amount emerged from the cluster groups of decoding, phonics and visual discrimination (N=4). A common thread in the *Journal of Educational Computing Research* was articles from international researchers in countries other than the United States. A study based in Italy, introduced a platform called En Plein, which is a motion based kinesthetic practice of phonological skills (Goffredo, Bernabucci, Lucarelli, Conforto, Schmid, Matilde Nera, Lopez, D'Alessio, & Grasselli, 2016). En Plein relies on the Microsoft Kinect motion sensor. A pre-test and post-test were conducted. The students who worked with En Plein showed improvements in their phonological awareness (mean increase of assessment scores of 9%), while their peers who received traditional instruction had a mean increase of 1%. The results pointed towards the indication that students were more confident in manipulating phonological units and increased their awareness of how words were built (Goffredo, Bernabucci, Lucarelli, Conforto, Schmid, Matilde Nera, Lopez, D'Alessio, & Grasselli, 2016).

Core themes for the Journal of Educational Computing Research. In total 22 clusters of meaning were identified within the articles pulled from 2007-2017 related to my topic. In the *Journal of Educational Computing Research* there were 13 themes that emerged from the articles. As I reviewed the 10-year span of the table of contents, I identified articles to review, read each selected article for clusters of meaning, I began to see patterns forming and clusters of themes developing. I was looking to determine if there was an overall connection between computer programming/coding and the skills needed to learn how to read and write.

Journal 4: *TechTrends.* An examination of *TechTrends* leads to 41 clusters of meaning as identified on Table 7 of Appendix E. Those topics included; (a) computing/programming/coding, N=6; (b), literacy, N=6; (c) reading and reading comprehension, N=6; (d) connected learning, N=4; (e) writing and digital writing, N=4; (f) communication skills, N=3; (g) sequencing, N=3; (h) technology integration, N=3; (I) vocabulary, N=3; (J) reading fluency, N=2; and (k) robotics, N=1. In the last step I synthesized the individual cluster groups that combined to create the overall clusters and those included, reading, reading comprehension, fluency and literacy, N=14; writing, digital writing, sequencing, vocabulary, and communication skills, N=13; computing, programming, coding and robotics, N=7, connected learning and technology integration, N=7. Within *TechTrends* a wide variety of topics were included; articles about technology education, educational trends, social networks, networking skills, strategies for teaching programming/coding, computer science in general, and applications for technology. *TechTrends* also incorporated articles related to teaching a variety of learners such as students with learning differences, students whose first language is a language other than English, and all grade level of students. Additionally, *TechTrends* incorporated studies on how to best integrate technology into the classroom with the best ethical and safety standards for students.

Cluster Themes for Journal 4

Reading, reading comprehension, reading fluency and literacy . Investigating the progression of reading, reading comprehension, reading fluency and literacy (N=14) was the most common cluster of meaning. Reading comprehension, or understanding the author's message, is a critical skill presented in the elementary grades. A study recorded in *Tech Trends* suggested that student-centered reading comprehension activities on the iPad can lead to better student achievement and motivation in reading (Moon, Wold, & Francom, 2016). In addition, Sessions, Kang & Womack (2016) conducted a study on improving writing instruction through iPad apps. This study reported that students wrote more cohesive, sequential stories using iPad apps verses traditional paper and pencil. Using the iPad apps also had an impact on motivation to write and made the writing process in general more social and engaging in the classroom.

In another study included in *TechTrends,* Russell and Cuevas, (2014) designed on line reading modules for a high school literature classroom. The modules were designed in partnership with the instructional technologist and the high school literature teacher to promote reading comprehension through modules that provided visual displays of text on the computer screen along with cognitive tools. Many high school students do not have the higher order thinking skills necessary to comprehend advanced level literature (Alfassi, 2004). The modules in this study were created to help students with comprehending advanced level texts, and they were found to boost comprehension of specific content and over time, even improved overall reading skills (Russell & Cuevas, 2014).

Writing, digital writing, sequencing, vocabulary, and communication skills. The second largest theme emerged by combining the cluster groups of writing, digital writing, sequencing, vocabulary, and communication skills, (N=13). In 2015, Darrington and Dousay published a study in *Tech Trends* that examined multimodal writing to help motivate struggling students to write. This study focused particularly on high school students. Within this study, change theory was explored to evaluate the benefits of multimodal works compared to traditional writing projects. The overall finding of the study was that multimodal writing was found to be more exciting for students than paper-based writing assignments (Darrington & Dousay, 2015). Writing involves developing sequencing, vocabulary and the ability to communicate ideas.

Computing, programming, coding and robotics. The theme with one of the fewest frequencies emerged from the cluster groups of computing, programming, coding, and robotics (N=7). Through my research, I discovered multiple studies that incorporated various reading and writing skills/concepts that can be accessed through coding/programming. Scouring the table of contents from 2007 to 2017, 23 articles referenced the connection to reading and writing. In Table 7 of Appendix E it is noted that seven articles were coded for significant words, clusters and themes that emerged from each one. Various skills were identified in the research. Reading skills include incorporating on line literature circles, improvement in reading comprehension, vocabulary development, fluency, and decoding skills (Boeglin-Quintana, & Donovan, 2013). In addition, Delacruz found that using Near Pods in elementary guided reading groups increased comprehension of the text presented, increased motivation to read, and students the ability to focus better on the text (2014).

Connected learning and technology integration. The final theme emerged from the cluster groups of connected learning and technology integration (N=7). The term computational thinking was recently identified as a key 21st century skill for all students (Yadav, Hong, & Stephenson, 2016). Computational thinking involves breaking down complex problems into more manageable smaller problems, using a sequence of steps (algorithms) to solve those problems, reviewing how the solution can be applied to solve other problems, and finally determining if a computer can assist in more efficiently solving problems (automation), (Wing, 2006). In a study conducted by Yadav, Hong, and Stephenson, 2016, and published in *Tech Trends,* it was suggested that computational thinking is a key skill that students need in order to move from being technology-literate to using technology and computational tools to solve problems.

Core themes for the journal TechTrends. In total 41 clusters of meaning were identified within the articles pulled from 2007-2017 related to my topic. In the journal

TechTrends, there were 11 themes that emerged throughout the articles examined. As I reviewed the 10-year span of the table of contents, I identified articles to review, read each selected article for clusters of meaning, and began to see patterns forming and clusters of themes developing. I was looking to determine if there was an overall connection between computer programming/coding and the skills needed to learn how to read and write. In total I identified 41 clusters of meaning and 11 themes. The theme that emerged consolidated into one overarching theme. *TechTrends* incorporates papers that emerge from technology conferences and events. The overarching theme for the journal *TechTrends,* truly is trends in research and education as it relates to technology.

Cycle 4: Summarizing the Composite Theories that Emerged from Data Source 1

The final coding cycle for the *Computers and Education*, *Educational Technology Research and Development*, *Journal of Educational Computing Research*, and *Tech Trends* involved reviewing significant statements, finding clusters of meaning, and lastly, finding the theories that evolved from the data. As reading and writing skills often intersect, the clusters of meaning and significant statements emerging from the scholarly journals also intersected. The core themes from each journal are presented in the next section.

Core Themes for Data source 1: Journal

Computers & Education. From the journal *Computers & Education*, the first overarching theme was preliminary skills necessary in order to acquire the ability to comprehend what a student is reading (vocabulary, literacy, reading, sequential analysis and writing). The second overarching theme that evolved was digital literacy (coding, programming, robotics, computational thinking, communication skills, and problem solving.) The *Computers & Education* journal seeks to promote not only the necessity of incorporating technology into the classroom, but also the research behind how becoming digitally literate can benefit students in other subject areas.

***Educational Technology Research and Development*.** The first overarching theme that I discovered from the 10-year span of articles within the journal *Educational Technology Research and Development* was cross-curriculum integration. Many of the articles discussed how coding/programming could be applied to a range of subject areas including language arts. The second overarching theme from this journal was the overall idea of technology literacy and how it can be applied to the development of a well-rounded learning program. This journal had many articles that specifically referenced scaffolded practice and cognitive development.

Journal of Educational Computing Research. I found that an overarching theme of the *Journal of Educational Computing Research,* was incorporating more interactive learning within the context of technology and digital learning. In total 22 clusters of meaning were identified within the identified articles pulled from 2007-2017 that related to my topic. Thirteen themes that emerged from within the articles examined. Multiple articles referenced not just technology, but interactive technology. Another overarching theme that I identified was the idea of cognitive interactive skills as referenced by the study on vocabulary development where a program defined words for a student, and the study in which, the computer program offered direct feedback to a student on his/her writing skill development.

TechTrends. The themes that emerged consolidated into one overarching theme. I view *TechTrends* as a journal that incorporated all topics related to technology developments in teaching and learning. In total 41 clusters of meaning were identified within the articles pulled from 2007-2017 related to my topic. In the journal *TechTrends,* there were 11 themes that emerged from the articles examined. *TechTrends* also incorporates papers that emerge from technology conferences and events. The overarching theme for the journal *TechTrends,* truly was trends in research and education as it relates to technology.

Overarching Themes from Data Source 1: Journals

In order to narrow down the overarching themes from each journal to a summary of the themes from Data Source 1, I reviewed the themes from each journal by writing the themes on index cards and sorting them into categories. I then categorized the themes. After sorting once, then sorting again I determined the overarching ideas from each, I then determined the three overarching themes from the four scholarly journals. As reading and writing skills often intersect, the clusters of meaning and significant statements emerging from the scholarly journals also intersected. The following data was found from the analysis across all four journals. The significant words and clusters of meaning were, cognitive learning, computational thinking, problem solving and cross curriculum integration, N= 10; comprehension in reading as it relates to technology integration and writing development, N=9; coding/programming/digital literacy/robotics, N=7; and communication and cooperative learning, N=3. In the final analysis of the four journals in data source 1 and based on the data presented, the message that these four journals are saying is that many of the skills necessary to read and write such as decoding, sequencing, reading comprehension, and writing are also skills needed to learn how to code/program on the computer.

Data Analysis of Data Source 2: Curriculum Standards

Cycle 1: Sample selection of standards

My second data source, and the source in which I repeated the same three coding cycles consisted of computer science standards across a variety of resources. The three coding cycles for the collection of standards included reviewing for significant words, searching for clusters of meaning and theme statements that would develop into theories. The process I followed included the review of the comprehensive K-12 standards from each of the five resources. The criteria for selecting the five resources included; identifying the most recent set of standards between the years of 2007-2017, determining that the standards were developed by an organization that was based in the United States, and included the standards for kindergarten-12th grade. All sources selected met the pre-determined criteria. Across all examples the standards were divided up in grade level groups; (a) pre-kindergarten through second grade, (b) third grade through fifth grade, (c) sixth grade through eighth grade, and (d) grades ninth through twelfth. By selecting these five sources of standards, I completed the first cycle of coding.

Included in the standards reviewed were *K-12 United States Computer Science Framework*, *Computer Science For All*, *International Standards for Computer Science Educators*, *Common Core*, and *Computer Science Teachers Association Standards*. I selected standards based on those developed by organizations within the United States and those identified as professional groups. I reviewed standards from these sources for the years 2007-2017. Computer science standards were first established in the United States in correlation with the common core standards, prior to that computer science and coding were only included as an elective in the high school. The following standards were reviewed, but not included in the data analysis due to either the scope of the standards, the research behind the standards, or the date of the most recent update of the standards, *National Math and Science Initiative*, *Next Generation Science Standards* and *K-12 Framework for Computer Science Education*.

As indicated in Table 8 of Appendix F, the following 34 clusters of meaning were identified and assisted in determining the selection of my data sources because of the significant words. Those clusters included; (a) networks, N=7; (b) coding/programming, N=5; (c) collaborator, N=2; (d) communication, N=2; (e) computational thinker, N=2; (f) cross-curriculum integration, N=2; (g) data and analysis, N=2; (h) decoding, N=2; (I) digital citizen, N=2; (J) innovator designer, N=2; (k) predicting/inferring, N=2; (l) sequencing, N=2; and (m) writing, N=2. The standards that I did select had all of these clusters in common and that is why I chose the standards that I selected and analyzed.

Cycle 2: Selection procedures for individual standards analysis

To examine the individual standards, I created a matrix of key codes, to search for within the standards. Using the key words discovered in the Cycle 1 analysis, I narrowed down to the following words; coding, programming, reading, writing, comprehension, decoding, vocabulary development, fluency, sequencing skills, and systems of understanding. As I reviewed each set of standards, I searched for the concepts of coding/programming as a required standard in the curriculum, and I looked for any cross-curricular connections in the standards. Once these terms were categorized and recorded on the matrix, the second cycle of coding was complete.

Cycle 3: Individual curriculum standards analysis

Standards 1: *Computer Science Teachers Association.* The standards produced by the Computer Science Teachers Association from 2017 were organized in grade level groups; K-2, 3-5, 6-8, 9-10, and 11-12. In Table 8 of Appendix F, it is noted that the standards were coded for significant words, clusters, and themes that emerged from each set of standards. In each grade level category, the standard was broken down into concept, sub-concept, and practice. The categories consisted of; computing systems, networks and the internet, data and analysis, algorithms and programming, and impacts of computing. Many skills included in language arts concepts taught in the early grades coincide with skills that are incorporated into the standards that I reviewed.

The *Computer Science Teachers Association* Standards contained 120 pages of standards from kindergarten through 12th grade. The standards were separated into five categories and varied in number depending on the grade level. The concepts included based on their frequencies of occurrences: algorithms and programming, N=55; impacts of computing, N=21; computing systems, N=14; data and analysis, N= 15; networks and the Internet, N=13. The cluster groups that I identified in order to determine the themes were, algorithms and programming, N=55; impacts of computing and computing systems, N=35, data and analysis, N=15, networks and the Internet, N=13.

Cluster Themes for First Identified Set of Standards

Algorithms and programming. Algorithms and programming as a category of the standards appeared in all grade levels. The frequency count for algorithms and programming appearing in the standards from Computer Science Teachers Association was N=55. As I reviewed the standards for each grade level, algorithms and programming appeared multiple times at each grade level grouping. The overarching theme of in K-2 focused on sequencing and how important it is to understand that steps must take place in a particular sequence in order to work properly. The algorithms and programming standards expectations begin to develop more complexity in grades 3-5. Students are expected to create programs that use variables to store and modify data as well as programs that include sequences, events, loops, and conditionals. Students are expected to create and clearly name variables that represent different data types and perform operations on their values. In grades 9-10 standards address objectives that ask students to create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

Impacts of computing and computing systems. The next theme which emerged from the cluster groups of impacts of computing and computing systems (N=35). This exemplifies the frequency count of computing and computing systems in the *Computer Science Teachers Association* standards. Examples of this theme were found in standards that supported the beginning steps of learning computing systems and computing starts in the kindergarten through second grade standards. As part of the computing systems standards for 3-5 grades students are taught how computer software and hardware work together as a system to accomplish tasks, and they are expected to determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies. In grades 6-8 for the computing systems standards, students are required to recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices. As students enter high school the category of computing systems becomes more focused on explaining and comparing levels of abstraction and interactions between application software, system software, and hardware layers.

Networks and the Internet. Based on the frequency counts, the theme emerged from the cluster groups of networks and the Internet (N=13). Language in the standards such as passwords, the purpose of passwords and why we use strong passwords to protect devices and information from unauthorized access is taught in K-2 exemplifies the frequency count of networks and the Internet in the *Computer Science Teachers Association standards*. Additionally, language about the importance of keeping login information private, and how to log off of devices appropriately was included. In grades 3-5, for the category of networks and the Internet, students begin to model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembles at the destination. That theme continues in grades 6-8 where students are expected to explain how physical and digital security measures protect electronic information, as well as explain the role of protocols in transmitting data across networks and the Internet. In 11-12 grades the complexity of how issues impact network functionality (e.g., bandwidth, load, delay, and topology) are explored.

Data and analysis. The theme emerged from the cluster groups of data and analysis (N=15). This exemplifies the frequency count of data and analysis in the *Computer Science Teachers Association standards.* The basic steps on how to store, copy, search, retrieve, modify and delete information using a computing device are taught in K-2 grades. In the upper level elementary grade levels, students begin to organize and collect data visually to highlight relationships and support a claim, propose cause-and-effect relationships, predict outcomes, or communicate an idea. In 6-8 grades in the category of data and analysis, students are expected to represent data using multiple encoding schemes and use computational tools and transform data to make it more useful and reliable. Before students exit high school, they should be able to use data analysis tools and techniques to identify patterns in data representing complex systems and select data collection tools and techniques to generate data sets that support a claim or communicate information.

Core themes for Standards 1: Computer Science Teachers Association

The terms of frequency, algorithms and programing appeared the most across the standards, N=55. Algorithms and programming appeared in all grade level standards and increased with complexity from one grade to the next. One of the overall themes that emerged from analyzing *Computer Science Teachers Association's* standards was the idea of learning about the interworking's of a computer, a network, and how the skills necessary to program a computer are all intertwined into the field of computer science. The second overarching theme that emerged from the *Computer Science Teachers Association* standards was that all students need to learn computer science across all grade levels and age groups. In total 118 clusters of meaning were identified in the *Computer Science Teachers Association* standards.

Cluster Themes for Second Identified Set of Standards

Standards 2: The *Common Core Standards.* The *Common Core Standards* were developed by the National Governors Association (NGA) and the Council Chief State School Officers Association (CCSSO) of the United States. Table 8 (Appendix F), illustrates the standards coded for significant words, clusters, and themes that emerged from each set of standards. These two groups-initiated feedbacks from multiple professional, vetted teacher organizations, content experts, state leaders and the general public. The CCSSO and NGA specified that the standards are; (1) research and evidence based, (2) aligned with college and work expectations, (3) rigorous, and (4) internationally benchmarked. The standard was only included in the document if its mastery was essential for college and career readiness for the 21st century in a globally competitive society. The skills of conducting research and the ability to consume and produce various modes of media are embedded into every aspect of the *Common Core Standards*.

The standards are divided into two categories: college and career readiness standards and K-12 standards. In the K-12 Standards there is a section on digital literacy categories. Included in the digital literacy skills are the same verbs used in the language arts standards; create, edit and format text, evaluate, decompose, code, program, communicate ideas, and develop and evaluate multimedia presentations with proper citations and grammar to exchange ideas. These particular skills were extracted from the K-5th grade digital literacy categories since they coincide the best with developmental early literacy skills. The skill standards are divided into three categories and five sub categories. The main categories are;

1. Demonstrate proficiency in the use of computers and applications as well as an understanding of the concepts underlying hardware, software and connectivity.
2. Demonstrate the responsible use of technology and an understanding of ethics and safety issues in using electronic media at home, in school and in society.

3. Demonstrate the ability to use technology for research, critical thinking, decision making, communication and collaboration, creativity and innovation.

The categories consist of an understanding of; spreadsheets (tables/graphs and charts), multimedia and presentation tools, acceptable use, copyright and plagiarism, research and gathering information, and communication and collaboration. The standards are delineated within each grade level as either introduced, reinforced, mastered, or optional. Under the category for spreadsheet (tables/charts and graphs), N=17; multimedia and presentation tools results were as follows, N=13; acceptable use, copyright, and plagiarism, N=21; research and gathering information the following results were noted, N=28; and communication and collaboration; N=22. The overall clusters that formed the themes include, spreadsheets, N=17; media and presentation tools, N=13, acceptable use, copyright, and plagiarism, N=21, research and gathering information, N=28, and communication and collaboration, N=22.

Cluster Themes for Second Identified Set of Standards

Spreadsheet (tables/charts and graphs). The theme emerged from the cluster groups of spreadsheets which included tables, charts and graphs, (N=17). This exemplifies the frequency count of spreadsheet related standards within the Common Core. All skills/standards presented in this section build in complexity from one grade level to the next. Students should be able to identify and explain terms and concepts related to spreadsheets (i.e. cell, column, row, values, labels, chart graph), and also enter/edit data in spreadsheets and perform calculations using formulas. In addition, students are expected to be able to use spreadsheets and other applications to make predictions, solve problems and draw conclusions.

Multimedia and presentation tools. The theme emerged from the cluster groups of multimedia and presentation tools, (N=13). This exemplifies the frequency count of multimedia and presentation standards within the Common Core. The introduction of multimedia and presentation skills begins in kindergarten. Students are expected to create, edit and format text on a slide through grade 5. Beginning in 6^{th} grade students are expected to create a series of slides and organize them to present research or convey an idea. Throughout this building of skills, students are expected to copy and paste graphics, change the size and position of graphics on a slide, use painting and drawing tools/applications to create and edit work, and learn how to watch age appropriate videos on line.

Acceptable use, copyright and plagiarism . The theme emerged from the cluster groups of acceptable use, copyright and plagiarism, (N=21). This exemplifies the frequency count of acceptable use, copyright, and plagiarism standards within the Common Core. The concept of teaching acceptable use, copyright and plagiarism is also introduced in kindergarten. These standards all align under the topic of digital citizenship. By 6^{th} grader the focus, safety on the computers expands. Standards address cyberbullying and strategies for dealing with cyberbullying situations and recognizing and describing the potential risks and dangers associated with various forms of online communications.

Research and gathering information. The theme emerged from the cluster groups of research and gathering information, (N=28). This exemplifies the frequency count of research h and gathering information related standards within the Common Core. Research and gathering information standards are also introduced in kindergarten.

The standards address using age appropriate technologies to locate, collect, and organize content from media collection for specific purposes, and citing resources. Kindergarten students are also introduced to evaluating teacher-selected or self-selected Internet resources in terms of their usefulness for research. By the end of 5^{th} grade students should be able to perform basic searches on databases to locate information, use content specific technology tools to analyze and gather data, and also appropriately use web 2.0 tools to gather and share information.

Communication and collaboration. The theme emerged from the cluster groups of communication and collaboration, (N=22). This exemplifies the frequency count of communication and collaboration related standards within the Common Core. The category groups of standards for communication and collaboration are introduced in 1^{st} grade. Skills expected to be mastered by the end of 5^{th} grade include working collaboratively online with other students, using a variety of technologies to communicate and exchange ideas, and use teacher developed guidelines to evaluate multimedia presentations for organization, content, design, presentation and appropriateness of citations. In addition, students are expected to create projects that use text and various forms of graphics, audio, and video to communicate and exchange ideas.

Core themes for standards 2: Common Core State Standards K-12. Across all of the standards is the theme of communication and working together to accomplish presentation projects. In addition, the second overarching theme revolved around students being expected to present, organize, and analyze data. In terms of frequency, research and gathering information appeared the most in the *Common Core* Standards, N=28. In total 101 clusters of meaning were identified in the *Common Core* standards.

Standards 3: Code.org. *Code.org* was launched in 2013, by two brothers of Iranian-American descent, Ali Partovi and Hadi Partovi. It started as a non-profit group focused on making computer programming more accessible and creating a database of all computer science classrooms in the United States (*Code.org*, 2013). Today the organization is still non-profit and dedicated to expanding access to computer science in schools and increasing the participation by women and minorities in the field of computer science. The vision for *Code.org* is that every student in every school has the opportunity to learn computer science. In addition, *Code.org* increases diversity in computer science by reaching students of all backgrounds where they are- at their skill level, and in their schools. *Code.org* (2013) provides the leading curriculum for K-12 computer science in the largest school districts in the United States (*Code.org*, 2013). The standards are grouped in courses A-F; starting with kindergarten and advancing through 5th grade. The content of these courses ranges from approximately 12 lessons in A & B, to almost 20 lessons in course F, with additional lessons available on the *Code.org* website to support further learning of specific concepts. The standards across the courses A-F consisted of the following categories; digital citizenship, N=5; events, N=4; loops, N=4; sequencing, N=4; conditionals, N=3; copyright and creativity, N=2; nested loops, N=2; binary, N=1; debugging, N=1; functions, N=1; and while/until loops, N=1. In Table 8 (Appendix F) illustrates standards were coded for significant words, clusters, and themes that emerged from each set of standards. The clusters that emerged to form the themes from this set of standards include, digital citizenship, copyright and creativity, N=7, events, loops, nested loops, while/until loops, conditionals and binary, N=15, and sequencing, functions and debugging, N=6.

Cluster Themes for Third Identified Set of Standards

Digital citizenship, copyright and creativity. The theme emerged from the cluster groups of digital citizenship, copyright and creativity, (N=7). This exemplifies the frequency count of digital citizenship, copyright and creativity related standards within *Code.org*. In the *Code.org* standards digital citizenship appears in all five courses which span from kindergarten through fifth grade. Copyright is a critical concept that students need to learn at an early stage in their educational careers. This skill is directly taught in grades one and five through the *Code.org* standards.

Events, Loops, Nested Loops, While/Until Loops, Conditionals and Binary. The theme emerged from the cluster groups of events, nested loops, while/until loops, conditionals and binary, (N=15). This exemplifies the frequency count of events, nested loops, while/until loops, conditionals and binary related standards within *Code.org*. Events are considered to be those standards that are taught "unplugged", (not directly on the computer). The introduction of basic loops is located in the standards for kindergarten, first and second. In the upper level elementary standards students learn about nested loops and also requires them to use functions in new ways by combining them with while loops and if/then statements. Conditionals do not appear in the standards until grades 3, 4, and 5. The concept of conditionals refers to learning the implication of "while" loops, "until" loops, and "if/else" statements when creating code. In reference to binary images this means that students are taught how a computer can store complex information (such as photos and colors) in binary.

Sequencing, Functioning and Debugging. The theme emerged from the cluster groups of sequencing, functioning, and debugging, (N=6). This exemplifies the frequency count of sequencing, functioning and debugging related standards within *Code.org*. The skill of sequencing appears again in this Qualitative Content Analysis within every group of standards proposed by *Code.org.* Through natural patterns of development, sequencing is embedded in the computer science standards. Programming/coding requires a specific sequence in order to initiate a command. Students should be able to identify when a program is not working correctly and apply different algorithms to correct the problem. In addition, students must be able to recognize the functions of a program and understand how they are interconnected.

Core themes for standards 3: Code.org. The vision of *Code.org* is that every student in every school has the opportunity to learn computer science. In total 28 clusters of meaning were identified in the *Code.org* standards. The cluster of events, loops, nested loops, while/until loops, conditionals and binary, (N=15) appeared the most across the standards. The overarching theme that emerged from the *Code.org* standards was for students to be able to manipulate the programming of a computer through the necessary skills required.

Standards 4: *K-12 Computer Science Framework.* In the United States, the K-12 Computer Science Framework published by the federal government provides a broader view. This document is organized by digital literacies expected to be mastered by the end of 2nd grade, by the end of 5th grade, by the end of 8th grade and finally, by the time a student graduates from high school. For this framework, the standards are referred to as concepts and those concepts are separated into categories: computing systems, networks and the internet, data and analysis, algorithms and programming, and impacts of computing. As with the other standards reviewed, many of the concepts coincide with the necessary steps in reading and writing proficiency. Table 8 (Appendix F) illustrates the standards were coded for significant words, clusters, and themes that emerged from each set of standards.

Also included in the United States K-12 Computer Science Framework, are requirements for students to learn the concepts of system relationships and inferencing. In previous standards those skills are assumed, but they are directly identified in the K-12 Computer Science Framework. The clusters of meaning for the US K-12 Computer Science Framework include, algorithms and programming, N=20; data and analysis, N=16; computing systems, N=12; impacts of computing N=12 and networks and the internet, N=8. The core practices consist of:

1. Fostering an inclusive computing culture
2. Collaborating around computing
3. Recognizing and defining computational problems
4. Developing and using abstractions
5. Creating computational artifacts
6. Testing and refining computational artifacts
7. Communicating about computing

The concepts/standards are broken down by expectations for the end of each grade level beginning in second grade. All five overarching concepts are included at various levels of complexity by the end of 2nd grade, by the end of 5th grade, by the end of 8th grade and by the end of 12th grade.

Cluster Themes for Fourth Identified Set of Standards

Algorithms and programming. Learning about algorithms and programming (N=20) was the most common cluster of meaning extracted from the *K-12 Computer Science Framework.* This section of the standards is divided into five categories; algorithms, variables, control, modularity, and program development. The basic understanding of algorithms begins in 2nd grade, students are presented with the concept that people follow and create processes as part of daily life, and many of these processes can be expressed as algorithms that computers can follow. That concept extends to 12th grade where students are expected to comprehend that people evaluate and select algorithms based on performance, reusability, and ease of implementation.

Data and analysis. The second largest theme emerged from the cluster groups of data and analysis (N=16). Data and analysis standards include four sub concepts; collection of data, storage of data, visualization and transformation, and inference and models. The skills learned under this heading are extensive. For collection students begin learning that everyday digital devices collect and display data over time and that the collection and use of data about individuals and the world around them is a routine part of life. That understanding of collection extends until students get to 12th grade and they are expected to understand that data is constantly collected or generated through automated processes that are not always evident, raising privacy concerns.

Computing systems. Through the analysis process, the second largest theme discovered indicated that an important skill required is to have knowledge of computing systems (N=12). An example of this theme can be found in all grade levels. By the end of second grade, students are expected to understand computing systems. In 5th grade students are required to understand how computing devices may be connected to other devices or components to extend their capabilities and also understand that these devices can take many forms, such as physical or wireless. Students should know that hardware and software work together as a system to accomplish tasks, such as sending, receiving, processing and storing units of information as bits. The complexity of computing devices and how they are often integrated with other systems, including biological, mechanical, and social systems is an expected objective to achieve by the end of 12^{th} grade.

Impacts of computing. The theme of impacts of computing yielded a frequent cluster of meaning (N=12). The three concepts addressed in the standards about the impacts of computing include; culture, social interactions, and safety, law, and ethics. In 2^{nd} grade students learn that computing technology has positively and negatively changed the way people live and work. Computing devices can be used for entertainment and as productivity tools, and they can affect relationships and lifestyles. The possibility of harmful behaviors, such as sharing private information and interacting with strangers is also introduced. By the time students complete 12^{th} grade, students are looking at the design and use of computing technologies and how artifacts can improve, worsen, or maintain inequitable access to information and opportunities.

Networks and the Internet. The clusters of meaning with the fewest frequencies found in the *K-12 Computer Science Standards* included networks and the Internet (N=8). The two categories under the heading Networks and the Internet are network communication and organization and also cybersecurity. By the end of 2^{nd} grade students should understand how computer networks can be used to connect people to other people, places, information and ideas, and gain an understanding that the Internet enables people to connect with others worldwide through many different points of connection. Cybersecurity is introduced in 2nd grade and students learn that connecting devices to a network or the Internet provides a great benefit, but care must be taken to use authentication measures, such as strong passwords, to protect devices from unauthorized access. By the end of 5th grade students should know that information needs a physical path to travel to be sent and received, and that some paths are better than others. By the end of 12th grade students learn about network topology and how many devices can be supported.

Core themes for standards 4: K-12 Computer Science Framework. For this framework, the standards are referred to as concepts and those concepts are separated into sub-categories: computing systems, networks and the internet, data and analysis, algorithms and programming, and impacts of computing. The core concepts for the US K-12 Computer Science Framework includes; computing systems, networks and the Internet, data and analysis, algorithms and programming, and impacts of computing. The overarching theme for the K-12 Computer Science Framework is for students to be knowledgeable and competent in the interworking's of computers, computer systems and how algorithms and programming interact with the computer. In total 68 clusters of meaning were identified in the *K-12 Computer Science Framework* standards.

Standards 5: *International Society for Technology in Education.* The final set of standards reviewed were those developed by the International Society for Technology in Education (ISTE). The ISTE is a community of global educators who believe that technology has the power to transform teaching and learning, accelerate innovation, and solve problems in education (*ISTE.org,*1979). The vision for ISTE is that all educators are empowered with the skills necessary to accelerate innovation in teaching and learning and inspire learners to reach their greatest potential through incorporating technology into all classrooms. The standards identified by ISTE include empowered learner, digital citizen, knowledge constructor, innovative designer, computational thinker, creative communicator, and global collaborator. ISTE has a different vision for technology in education. Based on a frequency count, the overall standards divided into 6 categories and threaded through all of the clusters of meaning are communication and collaboration, N=10; critical thinking, problem solving, and decision making, N=10, research and information fluency, N=7; creativity and innovation, N=5, digital citizenship, N=5; and technology operations and concepts, N=4. Table 8 (Appendix F) illustrated that the standards were coded for significant words, and clusters, of meaning. In total there were 41 clusters of meaning identified in the ISTE standards. The standards from ISTE are not divided by grade levels like the others. The ISTE standards are intended to guide curriculum mapping, lesson design, professional learning, school technology planning, school improvement planning, teacher preparation, and national policy. These standards also are intended to correlate with the United States *Common Core* plan.

Cluster Themes for Fifth Identified Set of Standards

Communication and collaboration. One of the most common clusters of meaning extracted from the Common Core standards was communication and collaboration (N=10). A common thread identified throughout the communication and collaboration thread encourages the development of students' skills to build networks and customize their learning environments to support the learning process. Students are also expected to communicate complex ideas clearly and effectively by creating or using a variety of digital objects such as visualizations, models or simulations. The goal is for students to publish or present content that customizes the message and medium for a variety of audiences. In the area of communication, students should be able to contribute constructively to project teams, assume various roles and responsibilities to work effectively towards a group goal.

Critical thinking, problem solving, and decision making. The other most common theme emerged from the cluster groups of critical thinking, problem solving, and decision making (N=10). Within the category of critical thinking, problem solving and decision making the skills revolve around becoming an innovative designer, computational thinker, and global collaborator. The theme of innovative designer represents what should know and use to design a process for generating ideas, testing theories, creating innovative artifacts and/or solving authentic problems. Students should also be able to select and use digital tools to plan and manage a design process that considers design constraints and calculates risks, as well as develop, test and refine prototypes as part of a cyclical design process. Under the topic of computational thinker, students are required to formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.

Research and information fluency. The cluster theme of research and information fluency (N=7) are part of the expectation to be constructors of knowledge, computational thinkers, and creative communicators. Expectations for students are for them to be able to plan and employ effective research strategies to locate information and other resources for their intellectual and creative pursuits. In addition, students are expected to evaluate the accuracy, perspective, credibility and relevance of information, media, data and/or other resources. Students should curate information from digital resources using a variety of tools and methods to create collections of artifacts that demonstrate meaningful connections and conclusions.

Creativity and innovation. Creativity and innovation (N=5) theme spans across the themes of innovative designer and creative communicator. Students should know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts and solving authentic problems. In addition, students should communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate for their goals.

Digital citizenship. One of the clusters of meaning with the fewest frequencies found included digital citizenship (N=5). In the field of digital citizenship, the idea is for students to be able to articulate and set personal learning goals, develop strategies leveraging technology to achieve them, and reflect on the learning process itself to improve learning outcomes. Using technology to seek feedback that informs and improves their practice and also demonstrates their learning in a variety of ways is another expectation of digital citizenship.

Technology operations and concepts. The cluster of meaning with the fewest frequencies was technology operations and concepts (N=4). Technology operations and concepts consists of students knowing and understanding the fundamental concepts of technology operations, demonstrating the ability to choose, use and troubleshoot current technologies, and transferring their knowledge to explore emerging technologies.

Core themes for standards 5: International Society for Technology in Education. The standards identified by ISTE include empowered learner, digital citizen, knowledge constructor, innovative designer, computational thinker, creative communicator, and global collaborator. ISTE has a different vision for technology in education. At their core, the ISTE standards are about pedagogy, not tools. The standards emphasize the ways that technology can be used to amplify and transform teaching and learning. The ISTE standards focus on skills that support students as they become agentic, future-focused and adaptable. In total 41 clusters of meaning were identified from the ISTE standards, N=10; critical thinking, problem solving, and decision making, N=10, research and information fluency, N=7; creativity and innovation, N=5, digital citizenship, N=5; and technology operations and concepts, N=4.

Cycle 4: Summarizing the Composite Theories that Emerged from Data Source 2.

I reviewed five sets of standards from different sources all developed in the United States. Multiple commonalities emerged across all frameworks. Reading, writing, and coding/programming have similar developmental steps. In order to read and write an individual must be able to identify letters, symbols, and sounds of letters. The same method exists for coding/programming on the computer, an individual must be able to identify needed lines of code and how to order the letters and symbols in order to make the computer "do" what the programmer wants it to do. Presented below is a review of the core themes findings for each set of standards.

Computer Science Teachers Association Standards. The terms of frequency, algorithms and programing appeared the most across the standards, N=55. Algorithms and programming appeared in all grade level standards and increased with complexity from one grade to the next. One of the overall themes that emerged from analyzing *Computer Science Teachers Association's* standards was the idea of learning about the interworking's of a computer, a network, and how the skills necessary to program a computer are all intertwined into the field of computer science. The second overarching theme that emerged from the *Computer Science Teachers Association* standards was that all students need to learn computer science across all grade levels and age groups. In total 118 clusters of meaning were identified in the *Computer Science Teachers Association* standards.

***Common Core Standard*s**. The skills of conducting research and the ability to consume and produce various modes of media are embedded into every aspect of the *Common Core Standards*. The standards are divided into two categories: the college and career readiness standards and the K-12 standards. Across all of the standards is the theme of communication and working together to accomplish presentation projects. In addition, the second overarching theme revolved around students being expected to present, organize, and analyze data.

Code.org Standards. Across all of the standards is the theme of communication and working together to accomplish presentation projects. In addition, the second overarching theme revolved around students being expected to present, organize, and analyze data. In terms of frequency, research and gathering information appeared the most in the *Common Core* Standards, N=28. In total 101 clusters of meaning were identified in the *Common Core* standards.

K-12 Computer Science Standards Framework. For this framework, the standards are referred to as concepts and those concepts are separated into sub-categories:

computing systems, networks and the internet, data and analysis, algorithms and programming, and impacts of computing. The core concepts for the US K-12 Computer Science Framework includes; computing systems, networks and the Internet, data and analysis, algorithms and programming, and impacts of computing. The overarching theme for the K-12 Computer Science Framework is for students to be knowledgeable and competent in the interworking's of computers, computer systems and how algorithms and programming interact with the computer. In total 68 clusters of meaning were identified in the *K-12 Computer Science Framework* standards.

International Society for Technology in Education The standards identified by ISTE include empowered learner, digital citizen, knowledge constructor, innovative designer, computational thinker, creative communicator, and global collaborator. ISTE has a different vision for technology in education. At their core, ISTE standards are about pedagogy, not tools. The standards emphasize the ways that technology can be used to amplify and transform teaching and learning. The ISTE standards focus on skills that support students as they become agentic, future-focused and adaptable. In total 41 clusters of meaning were identified from the ISTE standards, N=10; critical thinking, problem solving, and decision making, N=10, research and information fluency, N=7; creativity and innovation, N=5, digital citizenship, N=5; and technology operations and concepts, N=4.

Overarching Themes from Data Source 2

In order to narrow down the overarching themes from each set of standards down to a summary of the themes from Data Source 2, I reviewed the themes from each set of standards by writing the themes on index cards and sorting them into categories. I then categorized the themes. After sorting once, then sorting again I had determined the overarching ideas from each, I then determined the five overarching themes from the five set of standards. The five overarching themes are; understanding the interworking's of computers, networks and skills necessary to program, students of all ages need to learn computer science, students need to learn how to design and present multimedia projects, and students need to learn how to organize, and analyze data. In analyzing the overarching themes from all five set of standards examined I discovered the following, interworking's of computers, N=5; coding/programming, N=5; presentations/communication skills, N=4; reading and writing skills, N=3; and analyzing data, N=3. In total there were 356 clusters of meaning identified in Data Source 2, standards. In the final analysis of the five sets of standards in Data Source 2 and based on the data presented, the message that these five sets of standards are saying is that many computer science skills are often interwoven with skills taught in other subject areas, for example, organizing and analyzing data, multimedia presentations, and basic computer skills.

Data Analysis of Data Source 3: Websites that Teach Coding/Programming

Cycle 1: Sample selection websites

For the third data source, websites that teach grade level students to code/program, were reviewed. My selection criteria were based on websites that were currently up and operating, websites that had a variety of methods for teaching coding, and also websites that were kid friendly as well as appropriate for older grade level students. Websites that used advanced language skills that required students to have a strong foundation in computer science vocabulary were not included in this analysis.

Also, websites that were difficult to navigate and access the coding lessons, were also eliminated. Most websites that teach coding are either geared towards children or adults. Before an individual can begin to access a website that teaches coding/programming, he must be able to search and find the website. I began to navigate through the instructions on each website for a basic beginner level coder. I selected websites for this study based on those that were for students in grade levels K-12. I also selected websites based on how inviting they were for children, for example; colorful, childlike animations, level of vocabulary in the lessons, and ease of accessing the lesson. The websites selected have been vetted in research and created for students.

Programming languages changed significantly between 2007-2017, therefore the websites that taught coding and were selected for inclusion in my study were addressed differently than the journals and standards that were selected for inclusion in the study. Not only have the programming languages themselves changed, but the way individuals learn how to code, where they learn how to code, and why they learn to code has changed. I could not go back and specifically dissect earlier websites because they no longer existed. The skill of teaching individuals the art of programming/coding through a website did not become popular until more recently, specifically within the last five years. My criteria for inclusion included websites that taught individuals of a variety of age levels and skill levels how to code, the ease of learning and implementing a coding exercise, and the ease of which the website was accessible and user friendly. I reviewed a total of eight websites that teach coding which included *Code Academy, Code Avengers, Code.org, Coursera.org, CSunplugged, edx.org, Kahn Academy, and 2 Simple to code.* I did not select *Code Avengers* due to the fact that in order to access the lessons a user was required to pay a $25.00 yearly subscription fee. I wanted to review coding websites that were free for students to access. I also considered including *Coursera.org* and *edx.org,* but when I explored the websites more closely, I determined that the lessons were for adult learners and college level students. The other excluded website from my original list was *Kahn Academy.* I chose not to include *Kahn Academy* because coding lessons represented a small section of the lessons on this website. The majority of the lessons were focused on teaching reading, writing, math and specific high school level subject courses. The Websites selected from the cycle 1 reduction process were *Code.org/ Hour of Code, Code Academy, 2 Simple to code, and CSunplugged* (Computer Science unplugged). I selected these four websites because they all focused exclusively on teaching coding skills and all of the lessons were at an appropriate instructional level for kindergarten through 12th grade students.

Cycle 2: Analysis Procedures of Websites

The three coding cycles for the websites included reviewing for significant words, searching for clusters of meaning, and theme statements that could develop into theories. In order to identify the words for coding, I, first, needed to determine the method that each site used to instruct students in learning to code. Since the third data source was websites, it was clear that the learner must have access to a computer as well as have basic reading and writing skills mastered in order to access one of the coding programs.

As I worked my way through each of the four websites, I noted the necessary preliminary skills required in order to progress through the modules which resulted in the codes used to direct the next cycle of analysis. In Table 9, (Appendix G), the clusters of meaning embedded within each the four websites are identified. The following significant words emerged from the reviewed websites resulting in a total of 29: (a) programming/coding, N=5; (b) sequencing, N=4; (c) decoding, N=3; (d) debugging, N=3; (e) writing, N=3; (f) inferencing, N=2; (g) predicting, N=2; (h) publishing, N=2; (I) reading, N=2; (J) web development, N=2; and (K) binary numbers, N=1.

The breakdown per website is as follows: *Code.org* website, N=8; *2SimpletoCode* website, N=8; *Code Academy,* N=7; and *CSunplugged,* N=6. The next section addresses the individual website analysis.

Cycle 3: Individual Website Analysis

Website 1: *Code.org*. The first website explored and the website with the largest number of clusters of meaning based on frequency was *Code.org* (N=91). This website offers a variety of coding lessons for students at various levels of expertise. First, there is a course labeled for pre-readers (ages 4-8). These lessons take students through videos where the act of coding and debugging are explained with animation. *Code.org* was launched in 2013, by two brothers of Iranian-American descent, Ali Partovi and Hadi Partovi. It was started as a non-profit group focused on making computer programming more accessible and creating a database of all computer science classrooms in the United States (*Code.org*, 2013). *Code.org* (2013) provides the leading curriculum for K-12 computer science in the largest school districts in the United States (*Code.org*, 2013). The standards are grouped in courses A-F, beginning at kindergarten and ending at the fifth grade. *Code.org* also includes Express courses, which are more advanced lessons on coding for ages individuals between the 9-18 years old. Specifically, on the *Code.org* website I reviewed the titles of the courses to determine clusters of meaning. Using the clusters of meaning frequency counts were recorded in a matrix. Due to the quantity of weblinks within each website selected lessons were analyzed using conventional Qualitative Content Analysis to find emerging clusters of meaning. Specifically, on the *Code.org* website I reviewed the titles of the courses to determine clusters of meaning. Within those titles I unveiled the following clusters of meaning; loops, N=28 ; sequencing, N= 19 ; digital citizenship, N= 10 ; conditionals, N= 7; sprites, N=7; events, N=6; functions, N=5; end of course project, N=4; binary, N=3; and impacts of computing, N=2. In Table 9 (Appendix G) it is noted that each website was coded for significant clusters of meaning, and themes that emerged from each set of lessons. The websites were so large and had multiple links and in order to interpret the data contained within each website I had to limit my review to the lessons that were easily accessible on the website. In total N=91 clusters of meaning extracted from the *Code.org* website.

Cluster Themes from Website 1: *Code.org*

Loops, sequencing, digital citizenship and conditionals. Loops (N=28) and sequencing (N=19). These were the two topics that appeared with the most frequency in the coding lessons on *Code.org*. The next most common topics that appeared in the *Code.org* website was digital citizenship (N=10) and conditionals (N=7).

Sprites, events, and functions. The second largest group of clusters of meaning included sprites (N=7), events (N=6), and functions (N=5). This second group refers to lessons about how programming works and the interworkings of programming languages.

End of course projects, binary numbers, and impacts of computing. This group represents the smallest number of clusters of meaning included in the coding lessons on *Code.org*. The smallest group consists of end of course projects, (N=4); binary numbers, (N=3); and impacts of computing, (N=2).

Core themes from website 1: Code.org . The overarching themes from website 1; *Code.org* include learning to program/code while producing a product such as a game, app, or picture. The second overarching theme is that computer science is a critical skill for students to learn and the opportunity should be provided for all. The total clusters identified from the *Code.org* website was N=91.

Website 2: *2simple.com.* The second website explored and the website with the next to the largest number of clusters of meaning based on frequency was 2 *simple.com, (N=57).* The website, *2simple.com* (N=57) is for beginning coders and readers. The creator of *2simple.com's* mission is to create a space where children can be creative while using technology and tools that introduce them to desktop publishing, graphing and coding. The company that created *2simple* is based in the United Kingdom. Today over 90% of UK primary schools use *2simple* software to implement simple coding lessons and programming to grade school students. *Purple Mash* is the website for younger students, and it is full of educational tools, educational games a themed resource to introduce students to programming and coding. Purple mash can be used as a class or for independent study. It also has a section on the website where teachers can track a student's progress on the courses. The goal of *2simple* is to take difficult concepts and make them accessible to children of all abilities. *2simple* also has software available for preschool children that introduces them to coding. Within the *2simple* to code website I found 57 clusters of meaning; 2 logo, (N=13), 2 blog, (N=8), 2 code, (N=8), 2 animate, (N=5); 2 make designs, (N=6), 2 text, (N=6); 2 calculate, (N=4); 2 chart, (N=4); and 2 paint, (N=3). Each cluster of meaning represents a group of lessons that were taught on the website and the number of lessons for each. I discovered the clusters of meaning by examining the courses offered on the website.

Cluster Themes from Website 2: *2 Simple to Code*

2 Logo, 2 blog, and 2 code. The largest number of clusters of meaning found in the website *2 simple.com* include, 2 logo, (N=13); 2 blog, (N=8); and 2 code, (N=8). The data represent the number of lessons for each topic that are on the website. There are three sections that learners can navigate through on this website. The first section by following the steps of a given code, children are able to create multi-media stories, paint pictures, or compose a multi-instrumental song. Then the student can begin learning the different commands of saving, printing, deleting, and also using a mouse. In the next section, titled “Code amazing programs” students are introduced to the beginning steps of learning to code. By following the steps of a given code students are able to create simple coding activities and gradually move up to more complicated tasks such as creating a game or app. In the next section, students are taught step-by-step directions on how to publish a product on the computer.

2 animate, 2 make (design), and 2 text. The second largest cluster group includes; 2 animate, (N=5); 2 make, (N=6); and 2 text, (N=6). In this section, students are guided through publishing projects. All three sections of this website require students to have the fundamental skills of reading, writing, sequencing, decoding, and predicting. The publishing projects section provides themed templates to create leaflets, postcards, brochures and other project type documents. While students are creating these pieces, they are working on reading skills, writing skills, spelling, grammar, and sequencing.

2 calculate, 2 chart and 2 paint. The last section on *2simple* provides activities for students to participate in games which also practices key math and English skills in a fun and motivating way. These clusters represent the smallest number of lessons on the *2 simple.com* website, 2 calculate, (N=4); 2 chart, (N=4); and 2 paint, (N=3).

Core themes from website 2: 2simple.com. The overarching theme for the *2simple* website is to teach students that through coding/programming they can produce a product either a document, a piece of art, a game, or an app. The second overarching theme would be to exemplify how reading/writing and coding are interrelated in the skills required to do each one. In total there were 57 clusters of meaning identified in the *2 simple.com* website.

Website 3: *Code Academy.com.* *CodeAcademy.com* exemplified N=46 clusters of meaning, which was the third highest number out of the websites explored. *Code Academy* is an education company. The company is committed to "disrupt" the current way that schools work by changing the way things work in the classroom and by bringing classrooms online. When someone first enters CodeAcademy.com, he will be asked to answer a series of questions which include his purpose for wanting to learn how to code (career, fun, school). *Code Academy* was created by Zach Sims and Ryan Bubinski in August of 2011. This website teaches coding to grade level students and continues with more difficult content for adult learners. The subject areas that are included in *Code Academy* include; web development, programming, data science, partnerships and design. There are multiple skill paths that you can enter on *Code Academy* including; analyze data with python, analyze data with SQL, create a back-end app with javascript, create a front end app with react, getting started with machine learning, and learn how to build a website. These are all different types of programming languages that can be applied to coding.

A learner can access the basic version of *Code Academy* for free and there is also an option to purchase a membership to *Code Academy pro.* For this analysis the 46 free lessons were included. For this analysis the topics of the lessons were used as clusters of meaning resulting in the following categories related to this study: computer science, (N=14); web development, (N=14); data science, (N=13); and code foundations, (N=5). These clusters of meaning appeared in each of the lessons on *Code Academy.* The link to these 46 lesson plans can be found on Table 9 (Appendix G).

Cluster Themes from Website 3: *Code Academy*

Web development and computer science. The two largest clusters of meaning represented on the *Code Academy* website include, web development, (N=14); and computer science, (N=14). Through the *Code Academy* courses one can learn the steps necessary in order to design the simplest or the most elaborate webpage. Courses offered include; Python, Javascript, and Vue.js. Students not only learn how to create a website but also how to correct errors and debug, how to test for web development issues, navigation design, and design techniques. There is also a course on how to deploy a website onto the Internet.

Date science and code foundations. The second largest group and the final two clusters of meaning on the *Code Academy* website include; data science, (N=13); and code foundations, (N=5). The courses that are included in the data science section are all about data, how to organize data, utilize data and interpret data. Some examples would be learning statistics with Python, practical data cleaning, data analysis with Panda, data visualization in Python, learning about linear data structures, and learning about complex data structures and how to manipulate them.

Core themes from website 3: Code Academy.com. The overarching theme for the website *Code Academy* is the application of reading and cognitive skills in order to learn various coding languages and applying those coding languages to the web design and development. This website does not start out with basic programming, or the teaching the preliminary skills necessary in order to learn how to program/code. The *Code Academy* website had N=46 clusters of meaning as the third website explored, *Code Academy.*

Website 4: *Csunplugged.org.* The last website examined that teaches coding was Computer Science Unplugged (*csunplugged.org*). There were 46 total clusters of meaning found in the website *Csunplugged.org.* The primary goal of the Unplugged Project is to promote computer science and computing in general, to young people as an interesting, engaging, and intellectually stimulating discipline. *Csunplugged* is a collection of learning activities that teach computer science through engaging games and puzzles that use cards, string, crayons and a lot of physical movement. The main principles that distinguish *Csunplugged* are; no computer is required, real computer science concepts, learning by doing, no specialized equipment is required, and stand-alone activities. The language and type of lessons change based on the country that the user is logging in from, which is asked at the initial log in on to the website. The program is divided in to four sections of lessons; boost creativity, code amazing programs, inspire young writers, and engage with games. There are five cluster themes that I discovered on the *CSunplugged* website, binary numbers, (N=6); searching algorithms, (N=6); sorting networks, (N=4); kid bots, (N=4); and error detecting correction, (N=3). Each count for the clusters of meaning represents the number of lessons on each topic included on the website *CSunplugged.*

Cluster Themes from Website 4

Binary numbers and searching algorithms. Binary numbers, (N=6), and searching algorithms, (N=6) represents the largest number of clusters of meaning equally. In the lessons students are given challenges based on a few simple rules, and in the process of solving those challenges they uncover ideas on their own. For example, instructional materials, such as games and puzzles use; cards, string, crayons, and physical activity to teach computer science. With *Computer Science Unplugged*, learners do not need to have a computer.

Sorting networks, kidbots and error detecting correction. The clusters of meaning that represent the smaller number include, sorting networks, (N=4); kidbots, (N=4), and error detecting correction, (N=4). All three clusters have the same amount of lessons included. The learners must have basic knowledge of symbol/letter relationships, sequencing, decoding and debugging to progress to more difficult lessons. *Computer Science Unplugged* provides lesson plans, printables, and background knowledge for the teacher.

Core themes from website 4: Csunplugged.org. The overarching theme from *Csunplugged* is that learning computer science does not have to be done on a computer. Students can begin to learn the skills of programming, coding, error detection, debugging and the introduction to algorithms through hands on, very kinesthetic, memorable way. The total number of clusters of meaning discovered on the *CSunplugged* website N=23.

This website also represents with the fourth website examined and the one with the smallest number of clusters of meaning identified.

Core themes from all websites examined in Data Source 3. The overarching themes from website 1; *Code.org* (N=91) include learning to program/code while producing a product such as a game, app, or picture. The second overarching theme is that computer science is a critical skill for students to learn and the opportunity should be provided for all. The overarching theme for the *2simple* (N=57), website is to teach students that through coding/programming they can produce a product either a document, a piece of art, a game, or an app. The second overarching theme would be to exemplify how reading/writing and coding are interrelated in the skills required to do each one. The overarching theme for the website *Code Academy* (N=46), is the application of reading and cognitive skills in order to learn various coding languages and applying those coding languages to the web design and development. The overarching theme from *Csunplugged* (N=23) is that learning computer science does not have to be done on a computer. Students can begin to learn the skills of programming, coding, error detection, debugging and the introduction to algorithms through hands on, very kinesthetic, memorable way. In total there were 213 cluster themes identified from Data Source 3.

Cycle 4: Summarizing the Composite Theories that Emerged from Data Source 3

In total there were 217 clusters of meaning identified in Data Source 3: Websites. In analyzing the overarching themes from all four websites, I discovered the following: coding/programming, N=94; reading and writing skills, N=53; analyzing data, N=38; and presentations/communication skills, N=32. Based on the analysis, each website examined presented learning to code in a different format with various themes. All of the websites taught the learner how to code in small, sequential steps, and in a specific programming language. All websites provided online feedback to the learner as well as computer-based rewards and accolades. To summarize the overarching themes from all four websites explored in this study, all four recommend coding/programming as an important skill for students to learn. All four websites start with skills that students should already know then extend those skills exemplify how they are already integrated into programming/coding; such as sequencing, inferencing, decoding, reading and writing. The websites examined also show how an unfamiliar skill as coding/programming can be taught through multiple modalities. In the final analysis of the four websites in Data Source 3 and based on the data presented, the message that these four websites are saying is that many computer science skills can be taught cross curricular and can be integrated through multiple modalities.

Cycle 5: Summary of Core Themes from all Three Data Sources

The total number of clusters identified from all three data sources totaled N=266, after sorting into themes and then determining overall themes that evolved from the data sources combined. The conventional Qualitative Content Analysis method I followed to determine the overall clusters of meaning included taking the final paragraphs that summarized the overarching themes for each data source and copying it into a new document. Once all three were on the new document, using inductive analysis I highlighted overall clusters of meaning from each of the three Data Sources. Four overall themes emerged from the three Data Sources. As I was analyzing the overarching themes from each Data Source, I reduced down the individual data source themes into a matrix. I then determined the commonalities across the themes from each Data Source. Once I had determined the overall composite themes, I analyzed and tallied the clusters that fell into each category and recorded the results. Four overall themes emerged from the 266 total clusters.

The overall data analysis of four scholarly journals, five sets of computer science standards and four websites:

1. Coding/programming/digital literacy/sequencing, N=106
2. Reading and writing connections, N=65
3. Computer Science, cognitive learning, computational thinking, and problem solving, N=56
4. Communication/cooperative learning, N=39

Coding/programming/digital literacy/sequencing. The most frequent message that evolved from the overarching data analysis of all three Data Sources shows that coding/programming can be incorporated into other subject areas. The data provides an explanation that the skills necessary to learn how to code/program are interrelated to the skills needed to learn how to read and write. Those skills include sequencing, decoding, inferring, story comprehension, reading engagement, and vocabulary development. Included in Data Source 1 is an article from *Educational Technology Research and Development* by Zhou and Yadav (2017) that investigated the effects of multimedia story reading and questioning on children's literacy skills, including vocabulary, story comprehension and reading engagement. The results showed significant interaction of media and questioning on targeting vocabulary, story comprehension, and reading engagement. Included in all four of the computer science coding standards reviewed in Data Source 2, were coding and programming skills from kindergarten through 12th grade expectations. The standards taught increased in complexity from one grade level to the next and all standards incorporated coding/programming and coding languages. The main purpose of computer science is to teach programming and how to "talk" to a computer and make it execute the commands that the user wants the computer to perform. In Data Source 3, websites that teach coding, teaching a student how to code/program is the purpose of the website. On each of the websites examined the lessons presented taught students at various levels of competency programming languages and how to produce a product (for example, a published paper or picture, a computer game, or a website), by implementing programming steps through a computer.

Reading and writing connections. The next composite theme or message is that skills necessary to read and write are also skills necessary to code/program a computer. In Data Source 1, Journals, the articles examined exemplified the connection between reading, writing and coding through multiple modes. A study recorded in *Tech Trends* suggested that student-centered reading comprehension activities on the iPad can lead to better student achievement and motivation in reading (Moon, Wold, & Francom, 2016). In *Educational Technology Research and Developmen,* Ponce, Mayer, & Lopez's (2013) study explored the effectiveness of a computer based spatial learning strategy for improving reading comprehension and writing skills. The findings indicated that students who participated improved their reading and writing skills significantly more than the study group that did not apply the computer based spatial learning strategy. Data Source 2, computer science standards all had reading and writing skills in the standards, including but not limited to: inferencing, sequencing, decoding and writing skills. Data Source 3, websites require students to have basic reading and writing skills in order to access the coding lessons and those skills increase as the level of complexity in the lessons presented increase.

Computer Science, cognitive learning, computational thinking, and problem solving. Higher level thinking skills are necessary in order to learn how to code/program a computer. In an article written by Chen, Shen, Barth-Cohen, Jiang, Huang, and Eltoukhy published in *Computers & Education,* an instrument was created and administered that measured the computational thinking of fifth grade students as they learned coding in robotics and everyday reasoning abilities (2017). The authors determined that participating in a robotics curriculum improved students computational thinking skills. Critical thinking, problem solving, and decision making was the most common cluster group found in the *Common Core* standards. Within this category the skills revolved around becoming an innovative designer who uses digital tools for generating ideas, testing theories, creating innovative artifacts and/or solving authentic problems. In Data Source 3, websites, all of the lessons examined presented the necessity for applying higher level thinking skills through coding/programming.

Communication/cooperative learning . Communication and presentation skills are embedded in learning to code/program and learning to read and write. Communication skills work hand in hand with writing skills, an example was found in *Common Core* standards, which included communication skills and collaboration starting in 1st grade and continued through 12th grade. In Data Source 3, websites, the lessons were interactive where students often worked in groups to complete a coding project. Through writing and publishing students communicate. Digital literacy incorporates communication with the computer, with peer groups and interactive personas.

Coding/programming, reading, and writing all involve communicating and working with others in order to share an idea.

Composite/Core Message

Among the three Data Sources examined the themes of cross curricular skills dominated the analysis. In the final analysis of the four journals in data source 1 and based on the data presented, the message that these four journals are saying is that many of the skills necessary to read and write such as decoding, sequencing, reading comprehension, and writing are also skills needed to learn how to code/program on the computer. In the final analysis of the five sets of standards in Data Source 2 and based on the data presented, the message that these five sets of standards are saying is that many computer science skills are often interwoven with skills taught in other subject areas, for example, organizing and analyzing data, multimedia presentations, and basic computer skills. In the final analysis of the four websites in Data Source 3 and based on the data presented, the message that these four websites are saying is that many computer science skills can be taught cross curricular and can be integrated through multiple modalities. In order to code/program a student must be able to read, write, and comprehend the steps necessary to command the computer. In addition, coding/programming requires one to apply algorithms, patterns, data analysis, and sequences which are all computational thinking. Coding on the computer cannot occur unless the commands are written in a specific order, adapted to the specific programming language, and applied accurately. Reading, writing and coding have similar prerequisite and developing skills including: sequencing of letters and sounds, decoding sounds and the commands necessary for coding on the computer, understanding and deciphering patterns and how those patterns form to make a word or a command, writing letters in a particular order in order to form words and also to form lines of code, and higher level computational skills that lead to comprehension of content read and commands executed on the computer for coding. Coding/programming can be taught and learned by students in kindergarten through 12th grade. Coding/programming aligns with the steps of learning to read and write.

Summary

This is an emerging Grounded Theory study and this chapter presented the data from examining three data sources, including four scholarly articles, five coding/programming standards, and four websites that teach coding. I was searching for significant statements and clusters of meaning that appeared through the Qualitative Content Analysis of each data source then conducting another cycle of analysis by combining the overarching themes from all three data sources to identify common themes that emerged. Through this research, I sought to answer the following question:

1. What theory emerged that connected learning to code and coding literacy to the developmental stages of learning to read and write?
2. What are the theories that are shaping coding literacy?
3. Do the standards on learning to code and learning to read and write intersect and what theories emerged from the two?

In the next chapter I will present results, implications, outcomes, discussions, and limitations of this research study. The next chapter will contain a review of the results and how they connect to the related literature in the field. I will also discuss the limitations, and finish with my stance on the outcomes and implications of this study.

CHAPTER V

Discussion, Findings, Limitations, Recommendations for Future Studies, Implications and Conclusions

This study investigated the intersection of skills necessary for computer programming/coding and the developmental stages of learning to read and write. The three data sources included scholarly journals that contained articles on coding and literacy, curriculum standards from a variety of sources within the United States and selected current websites that teach coding to students. The methodology of this study was a content analysis with the framework of a content analysis. I conducted this study to understand the similarities between the skills needed to learn how to code on the computer and the skills needed to learn how to read and write. My reason for conducting this line of research was to use the findings in purposeful ways to encourage school districts to incorporate coding into their curriculums as a requirement across all grade levels, not just an elective in high school. Through an inductive content analysis framework, and a Grounded Theory lens I determined that coding literacy is similar in skills to reading and writing literacy.

In Chapter 1, I explained how coding/programming on the computer is now considered a literacy. There was also a case made to examine why coding/programming is a language of the future and explained the increased number of jobs that will be available over the next five to ten years in the field of computer programming. In addition, legislation was presented that indicated an increase of incorporated technology skills into the curriculum for K-12 grade students in the United States. In Chapter 2, I examined scholarly journals and research-based books that covered the topics of coding literacy, digital literacy, developmental steps of learning to read and write, and technology literacy. Major researchers who were referenced in this study include Vee, Papert, Vgotsky, Glaser, Charmaz, Gee, and Bers. Chapter 3 explained the methodology of my study and the theoretical framework. This is a content analysis. In Chapter 3, I explained the reasons behind selecting content analysis and also explained the steps that I would follow to conduct the study. I followed the defining components according to the steps developed by Urquhart (2013). I explained my role as a researcher, research sites, data collection procedures, and also established the three data sources for this study. In Chapter 4, I presented the data discovered through the research conducted on three data sources: scholarly journals, curriculum standards and coding websites. The overall findings from Chapter 4 included four themes: coding/programming/digital literacy/sequencing, reading and writing connections, computer science/cognitive learning/computational thinking/problem solving, and communication/cooperative learning.

Chapter five includes discussion of findings, limitations, implications, possible areas for future studies, and conclusions from the researcher. This chapter contains discussion and future research possibilities that extend the research conducted in this study to help answer the research questions:

1. What theory emerges that connects learning to code and coding literacy to the developmental stages of learning to read and write?
2. What are the theories that are shaping coding literacy?
3. Does the current data on learning to code and learning to read and write intersect and what theories emerge from the two?

This chapter summarizes the research findings via the theories that emerged from the content analysis of three data sources. This chapter includes a discussion of the major findings related to the skills necessary for learning how to code on a computer compared to the developmental steps of learning to read and write.

Discussion of Findings

Core themes. After analyzing the clusters of meaning from all three Data Sources the major findings from this study were that there is a connection between the skills needed to code on a computer and the skills necessary to learn how to read and write. Those skills that emerged from the data include:

1. coding/programming/digital literacy/sequencing,
2. Reading and writing connections
3. Computer science, cognitive learning, computational thinking and problem solving
4. Communication/cooperative learning.

Coding/Programming/Digital Literacy/Sequencing. The message that evolved from the overarching data analysis of all three Data Sources shows that coding/programming can be incorporated into other subject areas. It is evident from this theme; the direction of technology is advocating the importance of students to be taught how to code/program. Coding/programming on the computer is a skill that requires an individual to be able to read, write, sequence and create commands. In order for the computer to execute a command from the programmer the steps must be entered into the computer in a program language and the steps must be in order. This is just like reading and writing. In order for one to be able to read he/she must be able to decode the sequence of letter sounds and combinations that make words. The words must then be put in a logical order to create a complete sentence or idea. Regardless of the form that the words are coming from, rather it be a note on a piece of paper, a written book, or a story on the computer, the person reading it must be able to decipher the order of letters and how the letters are sequenced to form words. Once a student is able to decipher letters and sounds and how they form words, then sentences and stories can be written. As in computer coding/programming, a specific sequence of commands must be written in order for the computer to execute; in reading and writing a specific sequence must also be created. A story, whether it is being read or written, needs to have a comprehensible sequence of sentences, events, and paragraphs. Sequencing skills are a necessity for reading, writing and programming. The term digital literacy means any literacy that involves the use of the computer. Coding/programming is digital literacy. It is the ability to command a computer through a specific language to execute specific actions and sequences. When analyzing the standards, developers are setting curriculum expectations and the most frequent skills should include coding/programming and acquiring digital literacy skills. The theme of coding/programming/digital Literacy/sequencing emerged from through all data sources because it relates to the literacy of learning how to direct a computer's memory to perform tasks. In order to program/code on a computer, reading, writing and sequencing skills are necessary. This theme also indicates websites which are constantly developing to teach coding to adults and children are in alignment with the literature and standards.

Reading and Writing Connections. Overall the skills needed for coding, reading, and writing, do intersect and intertwine. The data provides an explanation that the skills necessary to learn how to code/program are interrelated to the skills needed to learn how to read and write. The literature that I explored, as well as the findings from this study, guides educators to encourage the inclusion of coding/programming into standards. The findings identified included multiple reading and writing skills that students acquire while learning to code/program. When children are learning to read and comprehend what they read, it is critical that they acquire sequencing skills. As the student reads a story, in order to comprehend the elements of the story, the sequence of events must be understood. Additionally, a student must learn how to decode the sequence of letters that make a word before he can read the word. Decoding letter sequences is a necessary skill in learning to read and developing writing skills. Reading and writing skills work in coordination with each other. In order to code a student must be able to write the sequence of commands and enter them into the computer. Specifically, designed websites are tools for teaching students how to code/program. The websites can be introduced into the everyday curriculum which would allow for teachers to build upon that learning through writing, publishing, and creating products. The letters and symbols specific to the coding language must be written in the exact order, or the computer will not "understand" what to do. Writing is an integral part of coding/programing because a coder needs to be able to "write" the line of code. Each coding language is different, but all languages include an aspect of writing and sequencing in order to execute loops. Therefore, writing is related to the skill of coding/programming due to the fact that you must comprehend the programming language and write commands in order to code.

Computer Science, Cognitive Learning, Computational Thinking, and Problem Solving. Computer Science is the study of computers. Cognitive learning is active and constructive learning. Computational thinking is critical thinking that involves problem analysis and decomposition, algorithmic thinking, algorithmic expression, abstraction and modeling. Problem solving is the process of solving solutions to complex issues. In order to code on the computer all four of these skills are necessary. Through my research and data analysis the act of applying problem solving and critical thinking skills emerged in all Data Sources. The ten-year span of literature reviewed included this third theme connecting the actual act of coding/programming to the cognitive thought processes required for computer science. The essence of this theme was to explain that coding/programming is an act that involves many areas of the brain function. By requiring higher level cognitive skills, coding/programming is now incorporated into various areas of the computer science standards. The act of coding/programming through computer science takes a specific skill set that an individual must apply to a programming language. Computer programming requires problem solving strategies and logical thinking (Wang and Hwang, 2017). It is through websites that teach coding that students are able to apply those higher-level cognitive skills in the classroom, as well as at home.

Communication/Cooperative Learning. Through computer coding and computer science students are afforded the opportunity to work together on projects particularly when publishing a project, presenting a production from a computer created program, and/or participating in robotics competitions. In today's workforce, it is a rarity that a person works in isolation. There are project teams, production teams, presentation teams, implementation teams, just to name a few. Therefore, the communication/cooperative learning skills that students acquire through coding are exemplified in studies conducted by researchers who look at how computer coding can extend to other skills and areas of study. Those studies are then examined by curriculum directors who write standards. Computer coding allows students to work collaboratively in groups as well as learn from each other. Students learn how to communicate with the computer often through a website, while coding/programming and publishing products. Students also develop presentation/communication skills when they produce a product through coding, such as three-dimensional presentations. Students can create games, new programs, and code a robot to complete a task. All of these incorporate activities where students work in groups and build communication skills.

Findings Related to Literature

Coding/Programming/Digital Literacy/Sequencing. Glister, (1997), defined digital literacy as an ability to apply information from a variety of digital sources. In correlation with my findings, he explained that digital literacy is being able to acquire and master ideas from technologies, it is not just keystroking. Through my data analysis, my results also lead to the belief that learning to code/program can extend an individual's literacy of reading and writing and apply it to digital literacy skills. Studyvin and Moninger (1986) support my finding as they argue that learning to code has been shown to increase reading skills, language mechanics, and reading comprehension. I also found that through my data analysis that these skills coincide with the skills of learning to code.

Reimagining the role of technology in education and incorporating coding into the curriculum is considered important (Bers, 2018; Clements, 1999; diSessa, 2001; Flannery & Bers; 2013). My study aligns with the findings of these studies, as it also explores the emerging theory of aligning digital literacy with traditional literacy. These studies specifically discussed how learning to code can impact sequencing skills, decoding, language development and vocabulary, problem solving and computational thinking. The data from this study corroborates with the theory of these researchers, through the analysis from the three Data Sources, which also exhibit examples of how the role of technology is changing in education.

Reading and Writing Connections. As I reviewed my data from Chapter 4, a constant theme emerged that connected reading and writing to the same skills necessary to program/code on the computer. Through multiple research studies, including this content analysis Grounded Theory, multiple theories emerged that connect learning to code/program and the developmental stages of learning to read and write (Barreto Vavassori Benitti, 2012; diSessa, 2001; Korat & Shamir, 2012; Savage, Erten, Abrami, Hipps, Comaskey, & van Lierop, 2010). The constant changes in technology, provides continuous opportunities for how, when, and where we engage in reading and writing (Papert, 1987). Kellogg, Whiteford, and Quinlan (2010) offer an example of support for this theme as they focused on the connection between writing and technology through an examination of whether an automated essay feedback system could improve the writing performance of college students. A struggle often noted by college professors is the weak writing skills of new college students. The findings of my study also extend the connection between the skills necessary to learn how to read and write, and the skills necessary to learn how to code on the computer. In my study, this theory that writing and technology is interrelated is extended as shown in the frequent emergence of articles related to coding and literacy, the standards examined that included writing, and the writing skills that are needed in order to learn how to code through a website. In this current study, reading and writing were interweaved within all three Data Sources.

While reviewing my findings in relationship to the literature an idea that emerged was the effects of technology on literacy skills. diSessa (2001) explained that the basis for literacy include structured subsystems with specific rules of operation, symbol sets, patterns of combination, conventions and means of interpretation. Those subsystems of language in terms of technology include inscription forms, game interfaces, contemplative browsing, and simulations and calculations. Considering all of these relationships, it is of the essence that we consider technology as a literacy. Many types of digital activities provide authentic and motivating contexts for developing specialized vocabulary. This vocabulary development and application of new vocabulary transfers over into comprehending higher level texts and writing more complex compositions and possible publications/presentations.

Coding as a literacy emerged from my data analysis. Judson (2009) argued that technology literacy gains could lead to heightened subject matter confidence and reflect improved ability to use technology as an avenue for new learning and application of that learning to other subject areas. Learning to code does not only impact the skill of making a computer do what you tell it to do. Coding by nature integrates multiple interdisciplinary skills that are important for students to learn to be successful citizens. Gee (2013) argued for well-designed games that build literacy skills, create problem-solving spaces with feedback and clear outcomes that lead to real, deep, and consequential learning. Many types of digital activities provide authentic and motivating contexts for developing specialized vocabulary that will be needed in the future. Coding apps are designed to encourage the creation of digital content rather than just consume it.

The apps provide an avenue where students must think creatively, reason systematically, and work collaboratively to code.

Computer Science, Cognitive Learning, Computational Thinking, and Problem Solving. Yadav, Hong, and Stephenson (2016) suggested that computational thinking is a key skill that students need in order to move from being technology-literate to using technology and computational tools to solve problems. An example of a study that supports this theme assessed the impact of meta-cognitive training on students' understanding of basic programming concepts (Cetin, Sendurur, E., & Sendurur, P., 2014). Their study investigated meta-cognitive training on students' programming achievement, retention of programming knowledge, and explore students' experiences and opinions related to meta-cognitive training. As the third highest theme according to frequency in the data analysis, technology can be viewed as a way to extend a student's learning.

Researchers who made a profound impact on the establishing the connection between coding/programming on the computer and literacy skills included Gee, (2013), Vee, (2013, 2017), Clements, (1999) and Bers & Horn, (2010), Bers, (2018). Researchers who have made the connection between coding and literacy are discovering more and more intersections between the two (Bers, 2018; Kafai & Burke, 2014, 2016; Papert, 1980). My current study extends the findings of these prominent researchers from the field in the emerging theory that connects digital literacy and coding to traditional literacy skills.

Clements (1999) visualized a future of educational computing research through a case of computer coding. Through my data analysis I found that computer coding reaches across multiple disciplines of learning. Clements (1999) also determined in his study where he examined how computer coding can be infused into multidisciplinary subjects including mathematics, problem solving, higher order thinking, language arts, creativity, and social-emotional development that learning to program increases a student's language skills (Lehrer & deBernard, 1987), readiness skills, visual discrimination, visual motor skills (Reimer, 1985), and an increase in first graders scores on assessments of visual motor development, language, and listening comprehension (Robinson, Gilley, & Uhlig, 1988; Robinson & Uhlig, 1988).

Communication/Cooperative Learning. Piaget (1977) described learning as the process of creating artifacts of personal and social relevance, connecting old knowledge to new knowledge, and interacting with others. Research supports the belief that collaborative learning is an effective strategy for teaching, by encouraging student interaction and offering a variance to the explanation of content. Although, students must be taught and monitored to ensure the productivity of collaborative learning, programming can provide opportunities for students to problem solve in groups and learn from each other (Wang and Hwang, 2017). In addition, research showed that students who participated in collaborative learning activities, had higher self-efficacy and a lower cognitive load than students working individually on the same activity (Wang and Hwang, 2017). In corroboration with Wang and Hwang (2017), the emerging theory of the current study, also supports the finding that coding/programming helps develop cooperative and communication skills in learners.

Summary of Findings Related to Literature

As I researched for my literature review, I discovered that the connection between literacy and computer programming and/or interacting with computers became more prevalent between 2013-2017. For the years 2007 to 2013, a majority of the literature referenced teachers and students learning to use technology. During this time period, teachers were utilizing technology for a reward, or an activity that students could interact with after they finished all of their "real" work (Savage, Erten, Abrami, Hipps, Comaskey, & van Lierop, 2010). In more recent years, from 2009 forward, researchers began to explore how programming and utilizing the computer for instruction could be beneficial to developing literacy skills in students (Korat, 2010; McKenney & Voogt, 2008). This is important to note, because this study was seeking to answer if there was a connection between learning to read and write and learning how to code/program on the computer. The reason it was important for me to study this topic, was because of the increasing demand for computer programmers in our society. As an educator, my most important responsibility is to prepare students to become successful, productive citizens. In order to do that, it is imperative that I understand the skills necessary for my students to succeed.

While conducting the literature review for this study, multiple themes emerged from the years 2007-2017. The overarching themes included: reimagining the role of technology in education, effects of technology on literacy skills, coding as a literacy, relationship between coding and learning to read and write, and coding standards. These themes closely aligned with the overall findings from my data analysis in Chapter 4.

Findings Related to Lens of Grounded Theory

Multiple theories surfaced from the three data sources through this content analysis study. One of questions guiding this study was to understand what theory emerges that connects learning to code to the developmental stages of learning to read and write. The second question was to uncover what theories are shaping coding literacy. Yes, through my study and data analysis my research questions were answered. There is a connection between the skills necessary to learn how to read and write (traditional literacy) and the skills needed to learn how to code/program a computer (digital literacy). Multiple theories have emerged that connect learning to code/program and the developmental stages of learning to read and write (Clements, 1999; Hutchinson, Nadolny & Estapa, 2015; Kazaoff & Bers, 2014; Kazakoff, Sullivan, & Bers, 2012; Papert, 1980; Rich, Browning, Perkins, Shoop, Yoshikawa, & Belikov, 2018; Snow, Tabors, Nicholson, & Kurland, 1994). This research does help us to better understand the theories emerging that connect learning to code to the developmental stages of learning to read and write as well as those shaping coding literacies. My research contributes to the growing number of studies that provide evidence of the connection between traditional literacy and digital literacy.

Sequencing, inferencing, and decoding skills are essential to both reading and writing and computer coding/programming. It is through decoding that students learn to segment words into syllables and pronounce the word. Likewise, it is through decoding that computer programmers analyze what commands to type into a computer to produce an action through code.

When reading or writing a story, the sequence of events plays a critical role in forming a comprehensible series of events. Also, when programming a computer, it is the sequence of the codes that create the figure, the animation, or tells the computer to do what it is told to do. If the commands are given in the incorrect sequence, then the computer does not perform the correct action. Similar to when writing or reading a story, if the sequence of events does not fit together then the story does not make sense.

Learning to code/program on a computer does have skills that align with the developmental steps of learning to read and write. In fact, there are multiple theories that emerged when reviewing the three data sources of scholarly journals, coding and computer science standards, and websites that teach an individual how-to code. In addition, there are cognitive skills that are attained that further develop and strengthen as students learn more complex objectives and progress through their educational careers.

To physically program a computer, one must be able to write the strand of commands and the order of commands in a specific programming language. Improper commands, commands that are out of order, and the use of the wrong programming language will invariably result in the computer's inability to "read" what the human wants the hard drive to activate and implement.

The theories that were identified included multiple reading and writing skills that students acquire while learning to code/program. When children are learning to read and comprehend what they read, it is critical that they acquire sequencing skills. As the student reads a story, in order to comprehend the elements of the story, the sequence of events must be understood. Additionally, a student must learn how to decode the sequence of letters that make a word before he can read the word. Decoding letter sequences is a necessary skill in learning to read and developing writing skills. To physically program a computer, one must be able to write the strand of commands and the order of commands in a specific programming language. Improper commands, commands that are out of order, and the use of the wrong programming language will invariably result in the computer's inability to "read" what the human wants the hard drive to activate and implement.

Stages of Reading and Writing Development

The final question guiding this study was to uncover if the current data on learning to code and learning to read and write intersect and what theories emerge from the two. The theories emerged were compared to the stages of developmental reading by Chall (1976), and the developmental stages of writing from Graves (1994). As reading and writing skills often intersect, the theories that emerged from the scholarly journals I examined also intersected. The theories that emerged that connect learning to code and coding literacy and the developmental stages of reading and writing include; development of reading comprehension skills, vocabulary development and language skills, sequencing skills in reading and writing development, advances in reasoning abilities, fluency in reading and writing, decoding skills, and overall systems of understanding. The answer to the third and final question was yes, the current data analyzed in this study does show that the skills necessary to learn how to code and to learn how to read and write intersect.

Visual discrimination and visual motor skills are also skills that are needed in order to read and write, as well as program a computer. Individuals need to be able to distinguish between the various letter formations and word patterns in order to decode a written language. Likewise, each programming language has its own set of patterns and commands that combine to create a sequence that speaks to the computer's hard drive. The programmer must be able to determine the steps necessary to make the computer do what he or she wants it to do.

The other theory that emerged from the three data sources explored the application of vocabulary development, syntax, and language mechanics. Just as a story or book must have a designated sequence of events, when a student is learning how to write a composition, essay, or scholarly article, the sentence structure, varied vocabulary, and flow of ideas are important elements. Regardless of an individual's language or purpose for writing, whether it be an English composition or a series of commands for a computer program, the language is specific and applicable to the project at hand. The formation of a sequence of letters and a specific order of words creates a message. This message can be put on paper or within the hard drive of a computer through a programming language.

Limitations

There were several limitations to this study. In selecting journals to review for articles, I only included scholarly, peer reviewed journals. I looked for journals that contained not only research on computer science and coding, but also how computer science was incorporated into K-12 grade schools. In addition, I wanted articles that focused on literacy. Literacy in the broader sense of not just reading and writing, but also digital literacies. I found more articles in journals related to technology than journals focused on traditional literacy. In addition, since countries in the world are at various stages of development economically and with technology, I decided to select only journals published in the United States. In doing so, I was more familiar with the standards and platforms utilized in the schools that the articles researched.

As a result, only four of the journals were included in Data Source 1. The journals included were; *Computers and Education*, *Journal of Educational Computing Research*, *Journal of Research and Technology in Education,* and *Tech Trends*. Thus, a broader schema of journals, including international publications, could provide research that would impact and help guide the alignment of skills attained through learning to code.

Next, only programming/coding/computer science standards from states and organizations in the United States were examined. Again, there are a multitude of countries that are further advanced in that coding standards are a required component of Kindergarten through 12th grade curriculums. Therefore, a broader study could include standards from other countries and educational systems. This could expand on the realm of advancement in technology and digital literacies across multiple cultures, languages and levels of economic development. The design and organization of the varied school systems in different countries could also be explored.

The third limitation relates to the websites that teach coding and their constant program languages and coding loop updates. In other words, what one can learn from Code.org today, may be totally different from what can be learned next month. Technology is always changing and advancing much faster than we are able to adapt. New program languages are developed and integrated into technology systems daily. There are some program languages that have stayed constant such as Java, JavaScript, and Python, but they are constantly being updated. In addition, I only reviewed the lessons that were easily accessible on each website and I did not purchase additional access to the lessons available only through exclusive access.

Included in this study is a limitation on the Grounded Theory lens. Due to the data sources being controlled and limited, an exhaustive review was not completed. I narrowed Data Source 1 down to four journals, limited Data Source 2 to five sets of standards, and only included four websites in Data Source 3. I focused on journals that had the most articles related to coding/programming and reading and writing developmental skills. When considering the standards to include, I studied the ones that were developed within the United States. The third Data Source, websites I included current websites that taught coding skills to K-12 grade students, but it was not an exhaustive inclusion because the number of websites that teach coding are increasing every single day.

Future Studies

There are many opportunities for future studies in the area of computer coding and its connection to the developmental steps of reading and writing. One example could be a longitudinal study that examines the differences in educational and career attainment between a group of students who received a K-12 computer science/coding curriculum verses a group of students who do not (education level completed, career, income level). An ethnographic study could also be completed on the reading proficiency level of students who receive coding instruction. I can foresee many literacy studies that would incorporate coding into the curriculum and compare the various level of mastery in skills such as: reading comprehension, reading fluency, vocabulary and language development, reasoning abilities, and writing skills. The current study could also be broken down into specific skills instead of looking at the overall correlation between reading and writing literacy and coding literacy.

In reading instruction specifically, research that examines any connection between learning to code/programming and the possibility of contributing to the academic reading improvement for students with reading difficulties and reading learning disabilities would be a valuable avenue to explore. Students with reading disabilities face the same challenges of being able to navigate through the digital world. Every student should have the opportunity to learn the skills necessary to lead a productive life. Even with the extensive amount of research on reading instruction, specific literacy instruction for students with reading disabilities has typically focused on learning sight words necessary for daily living and independence (Browder, Wakeman, Spooner, Ahlgrim-Delzell, & Algozzine, 2006). Studies on the effects of teaching digital literacy skills to students with reading disabilities would be a valuable area to extend this research.

Recently in the United States, there has been a movement focused on women's rights and equal opportunity advancement in their careers. The STEM workforce is critical to America's innovative capacity and global advancements in technology. However, despite making up half of the United States workforce and half of the college-educated workforce, women are vastly underrepresented in STEM fields and in rates of computer science degree completion. (Beede, Julian, Langdon, McKittrick, & Doms, 2011). In the general workforce, women in STEM-related jobs earn 33% more than comparable women in non-STEM-related jobs (Beede et al., 2011). Women remain under-represented in the science workforce, with the greatest disparities occurring in engineering computer science (National Girls Collaborative Project, 2018). Another research direction that focuses on women in computer science could be to look at the underrepresentation of minority women (National Girls Collaborative Project, 2018). Minority women represent only 1 in 20 of the scientists in STEM related jobs (National Girls Collaborative Project, 2018). Therefore, there is a plethora of studies that could be conducted related to the possible reasons why there is such a low incidence of females moving into the computer science fields of study and what can be done to encourage more females to pursue STEM-related jobs in kindergarten through 12^{th} grade.

Recently, several school districts in the United States have begun to incorporate coding not through a computer science course, but as part of their foreign language offerings. Instead of including coding/programming in the sciences, coding/programming is being considered a foreign language that can be added to a student’s language development. Along those same lines, another area of future research could be examining a potential connection between learning to code/program and learning languages other than English.

Implications

The term “being literate” has evolved over the last 20 years as technology has progressed and become more of an integral part of our lives. Many new literacies have emerged, including digital literacies. As the job market changes and expectations for how work is accomplished changes, the methods we use to prepare students to be successful and productive citizens must also change. According to recent trends in literature and research, computer coding/programming is becoming a prevalent skill to learn and incorporate into K-12 curriculum standards. It is this notion that inspired me to do this study.

Implications for Students

The implications from this research study could impact the education of students by encouraging more states to require computer science standards for all K-12 grade students, and as a result impact the number of students leaving college prepared to enter computer programming/coding professions would increase. Through this study it has been noted that to be successful in the present-day economic society, individuals should extend his or her literacy knowledge to computer programming and coding (diSessa, 2001; Engelhardt & Balanskat, 2015; Gee, 2013; Glister, 1997; Papert, 1980).

Implications for Teacher Preparation

This research study could also impact professional development and teacher preparation programs by requiring teachers to learn more about computer science in order to teach it to their students and seek out professional development. Learning to code/program aligns with the skills needed to become more proficient readers and writers, thinkers and problem solvers. In addition, as this study is discovered and read, it will hopefully lead to future studies on the connection between coding and reading and writing skills and why it is important to teach coding skills.

Conclusion

In this content analysis, the similarities between coding on the computer and the developmental stages of learning to read and write were examined. The research conducted in this study is important because computer coding/programming is a literacy that has become an important skill in our current society. This research will accelerate the argument that computer science and coding are skills that should be taught in the K-12 curriculums across the United States. Especially since the number of occupational computer science openings is projected to yield over 1 million jobs from 2014 to 2024 (Fayer, Lacey, & Watson, 2017). In addition, this research will also make a case for including coding and digital literacy into the teacher preparation programs.

At the beginning of this study, I was wanting to discover the theories that emerged when researching the steps in learning to code on the computer and the developmental steps in learning to read and write. At the conclusion of this study, the results indicate that there are many skills that co-exist between computer coding and reading and writing. I have found that reading and writing skills are aligned with the skills necessary to code/program a computer. Specifically, in order to code, you must be able to read and write. In addition, a computer programmer needs to be able to sequence, predict, inference, manipulate, and command the English language from the basic level to the most advanced. Among the three Data Sources examined the themes of cross curricular skills dominated the analysis. Across all Data Sources and the clusters of meaning that evolved from each Data Source, the analysis lead to the overall theme of answering the original research questions from this study which was to determine if there was an intersection between the skills necessary in learning to read and write and the skills required to learn how to code/program a computer.

The findings from this study may provide new insight and encouragement for states to require computer coding/programming as part of their K-12 standards. Additionally, the connection between reading and writing and learning to code could encourage teachers to pursue professional development on incorporating computer science into their classrooms because they can now understand the connection and similarities between learning to code and learning to read and write. Since coding is becoming a necessary skill for the job market in computer science, this study should encourage teachers to want to include coding into their daily instruction. Finally, this study may also motivate educators to focus their efforts on including female students in STEM related fields of study. Females should be contributing equally to the computer science fields of study and research. The exponential rise of the use of technology has impacted society, how society functions daily, and educators have a tremendous challenge to prepare students for the digital future. Digital literacy skills, including coding/programming are essential for students to learn in order to be prepared for the technologically-oriented workplace.

Researcher's Reflection.

Conducting this study was personally important to me because I am a principal that wants to provide the best education preparation for my students in order for them to have successful, productive lives as adults. I want my students to leave my school with the skills necessary to think critically, problem solve as individuals and in cohesive groups. Learning to program/code on the computer can lead to an infinite number of careers. In addition, this study shows that learning to code can be taught in conjunction with learning to read and write. Computational thinking and problem solving are skills that can be built upon and give students a foundation for more specific areas of study in college.

In addition, this study is important to me as an educator that believes all children need an opportunity to learn. Traditional paper and pencil in not how most children learn in this highly technological world. Children are inundated with technology in every aspect of their worlds, we need to teach them how to use technology to make this world a better place for their future success.

REFERENCES

ACARA. (2016). *The Australian curriculum. Retrieved from* http://www.austrialiancurriculum.edu.au/download/f10

Alfassi, M. (2004). Reading to learn: Effects of combined strategy instruction on high school students. *Journal of Educational Research,* 97(4), 171-184.

Altin, H. & Pedaste, M. (2013). Learning approaches to applying robotics in science education. *Journal of Baltic Science Education*, 12(3), 365-377.

Altin, H., Pedaste, M., & Aabloo, A. (2011). Educational robotics and inquiry learning: A pilot study in a web-based learning environment. *2011 IEEE 11th International Conference on Advanced Learning Technologies.* Doi:10.1109/icalt.2011.72

American Library Association. (2000).

Anderman, E. M., Anderman, L. H., & Griesinger, T. (1999). The relation of present and possible academic selves during early adolescence to grade point average and achievement goals. *The Elementary School Journal,* 100(1), 3-17. doi:10.1086/461940

Apple Technology Company. (1976).

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems,* 75, 661-670. doi.10.1016/j.robot.2015.10.008

Bakhtin, M. M., & Morris, P. (2009). *The Bakhtin Reader: Selected writings of Bakhtin, Medvedev, and Voloshinov.* Brantford, Ont: W. Ross MacDonald School Resource Services Library.

Bandura, A. (1997). *Self-efficacy: The exercise of control.* New York: W. H. Freeman.

Barakat, N. (2011). Balanced integration of theory and applications in teaching robotics. *The International Journal of Learning: Annual Review,* 18(1), 245-258. doi.10.18848/1447-9494/cgp/v18i01/4744

Barker, B. S., & Ansorge, J. (2007). Robotics as a means to increase achievement scores in an informal learning environment. *Journal of Research on Technology in Education,*39, 229-243. https://dx.doi.org/10.1080/15391523.2007.10782481

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, 48-54. http://dx.doi.org/10.1145/1929887.1929905

Barreto Vavassori Benitti, F. (2012). Exploring the educational potential of robotics in schools: A systemic review. *Computer & Education*, 58, 978-988.

Barrow, L., Markman, L., & Rouse, C. E. (2009). "Technology's edge: The educational benefits of computer-aided instruction." *American Economic Journal: Economic Policy*, 1(1), 52-74.

Bawden, D. (2001). Information and digital literacies: a review of concepts." *Journal of Documentation,* 57(2), 218-259. https://doi.org/10.1108/EUM0000000007083

Beede, D., Julian, T., Langdon, G., McKittrick, B. K., & Doms, M. (2011). Women in stem: A gender gap in innovation. U. S. Department of Commerce, Economics and Statistics Administration. http://ssrn.com/abstract=1964782

Bengtsson, M. (2016). How to plan and perform a qualitative study using content analysis. *Nursing Plus Open*, 2, 8-14, doi.10.1016/j.npls.2016.01.001

Bers, M, & Horn, M. (2010). High-tech Tots: Childhood in a digital world.

books.google.com

Bers, M. U. (2018). *Coding as a Playground: Programming and computational Thinking in the early childhood classroom.* New York, NY: Routledge.

Bers, M. U., Ponte, I., Juelich, C., Viera, A., & Schenker, J. (2002). Teachers are designers: Integrating robotics in early childhood educational *International Technology in Childhood Education Annual,* 2002(1), 123-145.

Bertram, V. (2016, October 04). Computer science (not just coding) needs a bigger role in STEM education. *EdTech: Focus on K-12. Retrieved from* http://edtechmagazine.com/k12/article/2016/10/computer-science-not-just-coding-needs-bigger-role-stem-education

Birks, M., & Mills, J. (2015). *Grounded Theory: A practical guide.* Los Angeles: Sage Publications.

Boeglin-Quintana, B., & Donovan, L. (2013). Story time using iPods: Using technology to reach all learners. *Tech Trends,* 57(6), 49-56.

Boote, D. N., & Beile, P. (2005). Scholars before researchers: on the centrality of the dissertation literature review in research preparation. *Educational Researcher*, 34(6), 3-15.doi:10.3102/0013189x034006003

Browder, D. M., Wakeman, S. Y., Spooner, F., Ahlgrim-Delzell, L., & Algozzinexya, B. (2006). Research on reading instruction for individuals with significant Cognitive disabilities. *Exceptional Children*, 72(4), 392-408.

Bulman, G. & Fairlie, R. W. (2016). "Technology and education: Computers, software, and the Internet." NBER working paper no. 22237, Cambridge, MA.

Burks, D. (2017, February 16). 5 reasons why coding is a skill you should learn in 2017|

CBC Life. Retrieved from https://www.cbc.ca/life/wellness/5-reasons-why-coding-is-a-skill-you-should-learn-in-2017-1.3987041

Burnard, P. (1991). A method of analyzing interview transcripts in qualitative research. *Nurse Education Today*, 11(6), 461-466. Doi:10.1016/0260-6917(91)90009-y

Cetin, I., Sendurur, E., & Sendurur, P. (2014). Assessing the impact of meta-cognitive training on students' understanding of introductory programming concepts. *Journal of Educational Computing Research,* 50(4), 507-524.

Chall, J. S. (1976). *Learning to read: The great debate.* San Francisco: MacGraw-Hill.

Charmaz, K. (2000). Teachings of Anselm Strauss: Remembrances and Reflections. *Sociological Perspectives.* Doi:10.2307/41888823

Charmaz, K. (2002). Tenets of Terror. *Qualitative Inquiry,* 8(2), 189-190. doi:10.1177/10778004008002015

Charmaz, K. (2006). *Constructing Grounded Theory.* Los Angeles: Sage Publications.

Charmaz, K. (2006). *Grounded Theory: Objectivist and constructivist methods.* Thousand Oaks, CA: Sage Publications.

Charmaz, K. (2014). *Constructing Grounded Theory.* Thousand Oaks, CA: Sage Publications.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education,* 109, 162-175.

Chitterji, A. (2018). *Innovation and American K-12 Education.* University of Chicago Press.

Cho, E., Lee, K., Cherniak, S., & Jung, S. E. (2017). Heterogeneous associations of second graders' learning in robotics class. *Technology, Knowledge and Learning,* 22(3), 65-483. doi:10.1007/s10758-017-9322-3

Christiansen, O. (2007). A simpler understanding of classic GT: How it is a fundamentally different methodology. *Grounded Theory Review,* 3(6). Retrieved from http://groundedtheoryreview.com/2007/06/30/1122/

Cihak, D. F., Smith, C. C., McMahon, D., & Kraiss, K. (2015). Incorporating Functional digital literacy skills as part of the curriculum for high school students with intellectual disability. *Education and Training in Autism and Developmental Disabilities,* 50(2), 155-171.

Clements, D. H. (1999). The future of educational computing research: The case of computer programming. *Information Technology in Childhood Education Annual,* 1991(1), 147-179.

Clements, D. H. (1987). Longitudinal study of the effects of logo programming on cognitive abilities and achievement. *Journal of Educational Computing Research,* 2, 73-94.

CodeAcademy.com. (2011). New York, NY. https://www.codecademy.com

CodeAvengers.com. (2019). New Zealand. https://www.codeavengers.com

Code.org (2013). Anybody can learn. Retrieved from https://code.org/

Common Core Standards. (2009). www.corestandards.org

Computer Science For All. (2018). https://www.csforall.org

Computer Science Teachers Association Standards. (2017). https://csteachers.org/page/standards

Connolly, T. M., Boyle, E. A., Macarthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers and Education,* 59(2), 661-686. doi:10.1016/j.compedu.2012.03.004

Constantinescu, A. (2007). Using technology to assist in vocabulary acquisition and reading comprehension. *The Internet TESL Journal,* 8(2). Retrieved from http://iteslj.org/Articles/Constantinescu-Vocabulary.html

CS4RI. (2016). *Computer Science for Rhode Island.* Retrieved from http://www.cs4ri.org/

CSTB, Computer Science Telecommunication Board. (2010). Report of a workshop on scope and nature of computational thinking. Washington, D. C. The National Academic Press.

CSUnplugged.com. https://csunplugged.org/en/

D'Agostino, J. V., Rodgers, E., Harmey, S., & Brownfield, K. (2015). Introducing an iPad app into literacy instruction for struggling readers: Teacher perceptions and student outcomes. *Journal of Early Childhood Literacy,* 16(4), 522-548. http://dx.doi.org.ezproxy.shsu.edu/10.1177/1468798415616853

Darrington, B. & Dousay, T. (2015). Using multimodal writing to motivate struggling students to write. *Tech Trends,* 59(6), 29-34.

Deckler, J., & Ifenthaler, D. (2017). Computational thinking as an interdisciplinary approach to computer science school curricula. In P. Rich & C. Hodges (Eds.), Emerging research, practice, and policy in computational thinking. Cham: Springer. https://doi.org/10.1007/978-3-319-52691-1_4

DelaCruz, S. (2014). Using nearpods in elementary guided reading groups. *Linking*

Research & Practice to Improve Learning, 58(5), 62-69.

Dewey, J. (1938). *Human nature and conduct, an introduction to social psychology.* New York: Allen.

diSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy.* Cambridge, MA: MIT Press.

Dockterman, E. (2014,). Google's made with code' kicks off with Mindy Kaling, Chelsea Clinton. Retrieved from http://time.com/2901899/google-made-with-code-girls-in-tech/

Domas White, M., & Marsh, E. E. (2006). Content analysis: A flexible methodology. *Library Trends,* 55(1), 22-45.

Dorian, N. C., & Barton, D. (1995). Literacy: An introduction to the ecology of written Language. *Language,* 71(3), 565. Doi:10.2307/416221

Durkin, D. (1981). Reading comprehension instruction in five basal reader series. *Reading Research Quarterly,* 16(4), 515-544.

EDX.org. https://www.edx.org

Elo, S., & Kyngas, H. (2008). The Qualitative Content Analysis process. *Journal of Advanced Nursing,* 62(1), 107-115. Doi:10.111/j.1365-2648.2007.04569.x

Engelhardt, K. & Balanskat, A. (2015). Computing our future. Computing programming and coding. Priorities, school curricula and initiatives across Europe. European School net.

Erdogan, N., Corlu, M. S., & Capraro, R. M. (2013). Defining innovation literacy: Do robotics programs help students develop innovation literacy skills? *International Journal of Educational Services,* 5(1), 1-9.

Every Student Succeeds Act (ESSA). (2015). U. S. Department of Education.

Fayer, S., Lacey, A., & Watson, A. (2017). STEM occupations: past, present and future. U.S. Bureau of Labor and Statistics.

Federal Communications Commission. (2010). Connecting America: The National Broadband Plan.

Fessakis, E. G., Gouli, E., & Macroudi, E. (2013). Problem solving by 5-6-year-old kindergarten children in a computer programming environment: A case study. *Computers & Education,* 63, 87-97.

Feurzeig, W., Bolt, B., & Newman. (1981). Microcomputers in education: Report no 4798. National Institute of Education, Washington, D. C.

Fitzgerald, S., Simon, B., & Thomas, L. (2005). Strategies that students use to trace code: An analysis based in Grounded Theory. In *Proceedings of the first international workshop on computing education research,* 69-80. New York: ACM Press.

Flannery, L. P., & Bers, M. U. (2013). Let's dance the "robot hokey-pokey!" *Journal of Research on Technology in Education,* 46(1), 81-101. doi:10.1080/15391523.2013.10782614

Fletcher, K. L., & Reese, E. (2005). Picture book reading with young children: A conceptual framework. *Developmental Review,* 25(1), 64-103.

Foreman, G. E., & Kuschner, D. S. (1983). *Piaget for teaching children: The child's construction of knowledge.* Washington, D. C., NAEYC.

Forsythe, G. E. (1959). The role of numerical analysis in an undergraduate program. *The American Mathematical Journal,* 66(8), 651-662.

Freire, P. (2017). The pedagogy of the oppressed, 50th edition. New York, NY: Bloomsbury Publishing Inc.

Furber, S. (2012). *Shut down or restart? The way forward for computing in UK schools.* London: The Royal Society.

Gardiner, B. (2014). Adding coding to the curriculum. *The New York Times*, 23rd March.

Gee, J. P. (2003). *What Video Games have to Teach us about Learning and Literacy.* New York, NY: Palgrave/Macmillan.

Gee, J. P. (2013). *The anti-education era: Creating smarter students through digital learning.* New York, NY: Palgrave/Macmillan.

Geist, E. (2016). Robots, programming and coding, oh my! *Childhood Education,* 92(4), 298-304. doi:10.1080/00094056.2016.1208008

Genishi, C., McCollum, P., & Strand, E. B. (1985). Research currents: The interactional richness of children's computer use. *Language Arts,* 62(5), 526-532.

Genlott, A. A., & Gronlund, A. (2013). Improving literacy skills through learning reading and writing: The iWTR method presented and tested. *Computers & Education,* 67, 98-104.

Glaser, B. G. (1965). *Awareness of Dying.* New York, NY: Aldine.

Glaser, B. G. (1965). The constant comparative method of qualitative analysis. *Social Problems,* 12(4), 436-445. doi:10.1525/sp.1965.12.4.03a00070

Glaser, B. G. & Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies For Qualitative Research.* New York, NY: Aldine.

Glaser, B. G. (1978). *Theoretical sensitivity: Advances in the methodology of Grounded*

Theory. Mill Valley, CA: The Sociology Press.

Glaser, B. G. (1992). *Basics of Grounded Theory analysis: Emergence vs forcing.* Mill Valley, CA: Sociology Press.

Glaser, B. G. (1998). *Doing Grounded Theory: Issues and Discussions.* Mill Valley, CA: Sociology Press.

Glaser, B. G. (2005). *The Grounded Theory perspective III: Theoretical Coding.* Mill Valley, CA: Sociology Press.

Glaser, B. G., & Strauss, A. L. (2015). Awareness of Dying. *Grounded Theory Review,* 14(2), 31-33.

Glister, P. (1997). *Digital Literacy*. John Wiley and Sons Inc.

Goals 2000: Educate America Act. (1994). 103d Congress, 1st session.

Goffredo, M., Bernabucci, C. L., Conforto, S., Schmid, M., Matilde Nera, M., Lopez, L., D'Alessio, & Grasselli, B. (2016). Evaluation of a motion-based platform for practicing phonological awareness of preschool children. *Journal of Educational Computing Research,* 54(5), 595-618.

Graves, D. H. (1994). *A fresh look at writing.* Toronto, Ont: Irwin Publishing.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the field. *Educational Researcher,* 42, 38-43. http://dx.doi.org/10.3102/0013189X12463051

Gruber, H. E., & Voneche, J. J. (1977). *The essential Piaget: An interpretive reference And Guide.* New York, NY: Basic Books Publishing.

Haas, C. (2009). *Writing technology studies on the materiality of literacy.* New York, NY: Routledge.

Hamilton, M. Barton, D., & Ivanic, R. (1994). Worlds of literacy. Bristol, United Kingdom: Multilingual Matters Ltd.

Harwood, T. G. & Garry, T. (2003). An overview of Qualitative Content Analysis. *The marketing Review,* 3(4), 479-498. doi:10.1362/146934703771910080

Hawisher, G. E. (1996). *Literacy, technology, and society: Confronting the issues.* Upper Saddle River, NJ: Prentice Hall.

Hawisher, G. E., & Selfie, C. L. (1999). *Passions, pedagogies, and 21st century technologies.* Logan: Utah State University Press.

Herold, B. (2016). Technology in Education: An overview. *Education Week,* Feb. 5, 2016. http://www.edweek.org/ew/issues/technology-ineducation/

Holdstein, D. H. & Selfe, C. L. (1990). *Computers and writing: Theory, research, practice.* New York: Modern Language Association of America.

Hsieh, H-F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research,* 15(9), 1277-1288.

Hutchinson, A. Nadolny, L., & Estapa A. (2015). Using coding apps to support literacy instruction and develop coding literacy. *The Reading Teacher,* 69(5), 493-503. doi:10.1002.trtr.1440

Importance of 21st Century Skills. (n.d.). *Creating Lifelong Learners: Using Project-Based Management to Teach 21st Century Skills,* 15-28. doi:10.4135/9781506313979.n3

International Society for Technology in Education. (1979). https://www.iste.org

International Standards for Computer Science Educators. (2016). https://www.iste.org/standards/for-students

Jenkins H. (2008). *Convergence culture: Where old and new media collide.* New York, NY: New York University Press.

Judson, E. (2009). Improving technology literacy: Does it open doors to traditional content? *Educational Technology Research and Development,* 58(3), 271-284. doi:10.1007/s11423-009-9135-8

K-12 Framework for Computer Science Education. (2016). https://k12cs.org

Kafai, Y. B., & Burke, Q. (2014). *Connected Code: Why children need to learn programming.* Cambridge, MA: The MIT Press.

Kafai, Y. B., & Burke, Q. (2016). *Connected gaming: What making video games can teach us about learning and literacy.* Cambridge, MA: The MIT Press.

Kahn Academy.com. https://www.khanacademy.org

Kazakoff, E. R., & Bers, M. U. (2014). Put your robot in, put your robot out: Sequencing through programming robots in early childhood. *Journal of Educational Computing Research*, 50(4), 553-573. doi:10.2190/ec.50.4.f

Kamil, M. L., Pearson, P., Moje, E. B., & Afflerbach, P. P. (2011). *Handbook of reading research volume IV.* New York, NY: Routledge.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2012). The effect of classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal,* 41(4), 245-255. doi:10.1007/s10643-012-0554-5

Kellogg, R. T., Whiteford, A. P., & Quinlan, T. (2010). Does automated feedback help students learn to write? *Journal of Educational Computing Research,* 42(2), 173-196.

Kemeny, J. G., & Kurtz, T. E. (1967). The Dartmouth time-sharing computing system. Final Report.

Khine, M. S. (2017). *Robotics and STEM Education: Redesigning the learning experience.* Cham: Springer International Publishing.

Knobel, M. & Lankshear, C. (2007). *A new literacies sampler.* New York, NY: P. Lang.

Knobel, M. & Lankshear, C. (2014). Studying new literacies. *Journal of Adolescent & Adult Literacy,* 58(2), 97-101. doi:10.1002/jaal.314.

Korat, O. (2010). Reading electronic books as a support for vocabulary, story comprehension and word reading in kindergarten and first grade. *Computers & Education,* 55, 24-31.

Korat, O., & Shamir, A. (2012). Direct and indirect teaching: Using E-books for supporting vocabulary, word reading, and story comprehension for young children. *Journal of Educational Computing Research,* 46(2), 135-152.

Krippendorff, K. (2004). *Qualitative Content Analysis: An introduction to its methodology.* Thousand Oaks, CA: Sage Publications.

Kucuk, S., & Sisman, B. (2017). Behavioral patterns of elementary students and teachers in one-to-one robotics instruction. *Computers & Education,* 111, 31-43.

Lankshear, C. & Knobel, M. (2008). *Digital literacies: Concepts, policies and practices.* New York, NY: Peter Lang.

Leech, N. L., & Onwuegbuzie, A. J. (2011). Beyond constant comparison qualitative data analysis: Using Nvivo. *School Psychology Quarterly,* 26(1), 70-84. doi:10.1037/a0022711

Lehrer, R., & deBarnard, A. (1994). Language of learning and language of computing: The perceptual-language model. *Journal of Educational Psychology,* 79, 41-48.

Leu, D. J. Jr., & Kinzer, C. K. (2000). The convergence of literacy instruction with Networked technologies for information and communication. *Reading Research Quarterly,* 35(1), 108-127.

Liao, Y. C., & Bright, G. W. (1991). Effects of computer programming on cognitive outcomes: A meta-analysis. *Journal of Educational Computing Research,* 7(3), 251-268. doi:10.2190/E53G-HH8K-AJRR-K69M

Lohr, S. (2015). *As tech booms, workers turn to coding for career change.* The New York Times, 28th July.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior,* 41, 51-61.

Lynch, T. L. (2017). *Strata and bones: Selected essays on education, technology, & Teaching English.* North Charleston, SC: CreateSpace.

Lysenko, L. V., & Abrami, P. C. (2014). Promoting reading comprehension with the use of technology. *Computers & Education,* 75, 162-172.

MacArthur, C. A., Graham, S., & Fitzgerald, J. (2006). *Handbook of writing research.* New York: Guilford Press.

Marshall, E. (2004). Stripping the wolf: Rethinking representations of gender in children's literature. *Reading Research Quarterly,* 39(3), 256-270.

Maxwell, J. A. (2013). *Qualitative research design: An interactive approach.* Thousand Oaks, CA: Sage Publications.

Mazzei, L. A., Jackson, A. & Youngblood, A. (2020). *Thinking with theory in qualitative research: Viewing data across multiple perspectives.* London and New York: Routledge.

McDonald, C. V. (2016). STEM Education: A review of the contribution of the disciplines of science, technology, engineering, and mathematics. *Science Education International,* 27(4), 530-569.

McKenney, S. & Voogt, J. (2008). Designing technology for emergent literacy: The PictoPal initiative. *Computers & Education,* 52, 719-729.

McLaughlin, Wallin, M. (1974). Evaluation and Reform: The Elementary and Secondary Act of 1965, Title I.

Mikropoulos, T. A., & Bellou, I. (2013). Educational robotics as mindtools. *Themes in Technology Science & Technology Education,* 6(1), 5-14.

Mioduser, D., Levy, S. T., & Talis, V. (2009). Episodes to scripts to rules: concrete-abstractions in kindergarten children's explanations of a robot's behavior. *International Journal of Technology and Design Education,* 19(1), 15-36. doi:10.1007/s10798-007-9040-6

Mills, J., Bonner, A., & Francis, K. (2006). The development of constructivist Grounded Theory. *International Journal of Qualitative Methods,* 5(1), 25-35. doi:10.1177/160940690600500103

Mills, K. A., Chandra, V., & Park, J. Y. (2013). The architecture of children's use of language and tools when problem solving collaboratively with robotics. *The Australian Educational Researcher,* 40(3), 315-337. doi:10.1007/s13384-013-0094-z

Mioduser, D., & Levy, S. (2010). Making sense by building sense: Kindergarten children's construction and understanding of adaptive robot behaviors. *International Journal of Computers for Mathematical Learning,* 15(2), 99-127.

Mishra, P., Terry, C. A., Henriksen, D., & The Deep Play Research Group. Square peg, round hole, good engineering. *Tech Trends,* 57(2), 22-25.

Moon, A. L. Wold, C. M., & Francom, G. M. (2017). Enhancing reading comprehension with student-centered iPad applications. *Tech Trends,* 61(2), 187-194.

Morse, J. M. (1991). Evaluating qualitative research. *Qualitative Health Research,* 1(3), 283-286. doi:10.1177/104973239100100301

Morse, J. M., & Richards, L. (2002). *README FIRST for a User's guide to Qualitative Methods.* Sage Publications.

Moss, B. (2008). The information text gap: The mismatch between non-narrative text types in basal readers and 2009 NAEP recommended guidelines. *Journal of Literacy Research,* 40, 201-219.

Nager, A., & Atkinson, R. D. (2016). *The case for improving U.S. computer science education.* Information Technology & Innovation Foundation. Retrieved from http://www2.itif.org/2016-computer-science-education.pdf

Nagy, W., & Scott, J. (2000). Vocabulary processing. In M. Kamil, M. Mosental, P. D. Pearson, & R. Barr (Eds.), *Handbook of reading research* (Vol. 3, 269-284). Mahwah, NJ: Lawrence Erlbaum.

National girls collaborative project. (2018). https://ngcproject.org

National Math and Science Initiative. https://www.nms.org

Next Generation Science Standards. https://www.nextgenscience.org

Neuendorf, K. A. (2017). *The Content Analysis guidebook.* Los Angeles, CA: Sage Publications.

New America.org. (1999). https://www.newamerica.org

No Child Left Behind (NCLB) Act of 2001, 20 U.S.C.A. 6301 et seq. West 2003.

Nofre, D., Priestley, M., & Alberts, G. (2014). When technology became language: The origins of the linguistic conception of computer programming, 1950-1960. *Technology and Culture,* 55(1), 40-75. doi: 10.1353/tech.2014.0031

Okita, S. Y. (2013). The relative merits of transparency: Investigating situations that support the use of robotics in developing student learning adaptability across virtual and physical computing platforms. *British Journal of Educational Technology,* 45(5), 844-862. doi:10.1111/bjet.12101

Onwuegbuzie, A. J., Leech, N. L., & Collins, K. M. (2011). Innovative qualitative data collection techniques for conducting literature reviews/research syntheses. In M. Williams & W.P. Vogt (*Eds.). The Sage handbook of innovation in social research methods, (pp.182-204).* Thousand Oaks, CA: Sage Publications, Inc. doi:10.4135/9781446268261.n13

Owston, R., Wideman, H., Sinitskaya Ronda, N., & Brown, C. (2009). Computer games development as a literacy activity. *Computers & Education,* 53, 977-989.

Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas.* New York: Basic books.

Papert, S. (1987). Computer criticism vs. technocentric thinking. *Educational Researcher,* 16(1), 22. doi:10.2307/1174251

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer.*

Basic Books: New York, NY.

Parsons, S. A. & Gallagher, M. A. (2016). A content analysis of nine literacy journals, 2009-2014. *Journal of Literacy Research,* 48(4), 476-502. doi:10.1177/1086296x16680053

Patton, M. Q. (1990). *Qualitative evaluation and research methods (2nd Ed.).* Newbury Park Calif: Sage Publications.

Patton, M. Q. (2002). Two decades of development in qualitative inquiry: A personal, experiential perspective. *Qualitative Social Work,* 1(3), 261-284.

Pea, R. D., & Kurland, D. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology,* 2(2), 137-168. doi:10.1016/0732-118(84)90018-7

Phillips, R. S., & Brooks, B. P. C. (2017). *The hour of code: Impact on attitudes towards and self-efficacy with computer science.* Retrieved from http://code.org/files/HourOfCodeImpactStudy_Jan2017.pdf

Piaget, J. (1977). *Psychology and epistemology: Towards a theory of knowledge.* New York City, New York: Penguin Books.

Ponce, H. R., Mayer, R. E., Lopez, M. J., & Loyola, M. S. (2018). When two computer-supported learning strategies are better than one: An eye-tracking study. *Computers & Education,* 125, 376-388.

Potocki, A., Ecalle, J. & Magnan, A. (2013). Effects of computer-assisted comprehension training in less skilled comprehends in second grade: A one-year follow-up study. *Computers & Education,* 63, 131-140.

President Trump's FY 2018 Budget. (2018). A new foundation for American greatness prioritizing students empowering parents. Retrieved from http://www.2.ed.gov

RaspberryPI.org, (2009). https://www.raspberrypi.org

Reimer, G. (1985). Effects of Logo programming experience on readiness for first grade, creativity, and self-concept. "A pilot in kindergarten." *AEDS Monitor,* 23(7-8), 8-12.

Reimagining the role of technology in education: 2017 National education technology plan update. (2017). U.S. Department of Education. Retrieved from http://tech.ed.gov

Reinking, D., Labbo, L. D., & McKenna, M. C. (2012). *Handbook of literacy and technology: Transformations in a post-typographic world.* Mahwah, NJ: L. Erlbaum Associates.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67. doi:10.1145/1592761.1592779

Resnick, M., & Seigel, D. (2015). A different approach to coding- How kids are making and remaking themselves from Scratch. *International Open Magazine.* Retrieved from http://internationalopenmagazine.org/2015-12-17-approach_to_coding.html

Resnick, M. (2006). Computer as paintbrush: Technology, play, and the creative society. In D. Singer, R. Golikoff, & K. Hirsh-Pasek (Eds.), *Play = learning: How play motivates and enhances children's cognitive and social-emotional growth.* New York, NY: Oxford University Press.

Ribeiro, R. (2013). Chicago makes computer science a core subject area. *EdTech Magazine.* Retrieved from http://www.edtechmagazine.com/k12/article/2013/12/chicage-makes-computer-science-a-core-subject

Rich, P. (2012). Inside the black box: Revealing the process in applying a Grounded Theory analysis. *The Qualitative Report,* 17(49), 1-23.

Rich, P., Browning, S., Perkins, M., Shoop, T., Yoshikawa, E., & Belikov, O. (2018). Coding in K-8: International trends in teaching elementary/primary computing. *Tech Trends,* https://doi.org/10.1007/s11528-018-0295-4

Robinson, M. A., Gilley, W. F., & Uhlig, G. E. (1988). The effects of guided discovery Logo on SAT performance of first grade students. *Education,* 109, 226-230.

Robinson, M. A., & Uhlig, G. E. (1988). The effects of guided discovery instruction on mathematical readiness and visual motor development in first grade students. *Journal of Human Behavior and Learning,* 5, 1-13.

Rodgers, E., Dagostino, J. V., Harmey, S. J., Kelly, R. H., & Brownfield, K. (2016). Examining the nature of scaffolding in an early literacy intervention. *Reading Research Quarterly,* 51(3), 345-360. doi:10.1002/rrq.142

Russell, R. & Cuevas, J. (2014). Designing customizable reading modules for a high school literature classroom. *Tech Trends,* 58(5), 71-80.

Saldaña, J., Saldaña, J., & Miles, M. B. (2015). *The coding manual for qualitative researchers qualitative data analysis: A methods sourcebook.* London: Sage Publications.

Savage, R. S., Erten, O., Abrami, P., Hipps, G., Comaskey, E., & van Lierop, D. (2010).

ABRACADABRA in the hands of teachers: The effectiveness of a web-based literacy intervention in grade 1 language arts program. *Computers & Education,* 55, 911-922.

Scaradozzi, D., Sorbi, L., Pedale, A., Valzano, M., & Vergine, C. (2015). Teaching robotics at the primary school: An innovative approach. *Procedia-Social Behavioral Sciences,* 174, 3838-3846. doi:10.1016/j.sbspro.2015.01.1122

Scherer, R., Siddiq, F., & Sanchez Viveros, B. (2018). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology.* Advance online publication. http://dx.doi.org/10.1037/edu0000314

Schmar-Dobler, E. (2003). Reading on the Internet: The link between literacy and Technology. *International Reading Association,* 47(1), 80-85.

Schreier, M. (2013). *Qualitative Content Analysis in practice.* London: Sage Publications.

Scribner, S. (1984). Literacy in three metaphors. *American Journal of Education,* 93(1), 6-21.

Selfe, C. L. (1999). *Technology and literacy in the twenty-first century: The importance of paying attention.* Carbondale: Southern Illinois University Press.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G. & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies,* 18(2), 351-380.

Sergeyev, A., Alaraje, N., Kuhl, Meyer, M., Kinney, M., & Highum, M. (2015).

Innovative curriculum model development in robotics education meet 21st century workforce needs. Presented at the American Society for Engineering Education Conference, Gulf Southwest-Midwest-North Midwest Sections.

Sessions, L., Kang, M. O., & Womack, S., (2016). The neglected 'R' improving writing instruction through iPad apps. *Tech Trends,* 60(3), 218-225.

Shamir, A., Korat, O., & Barbi, N. (2007). The effects of CD-ROM storybook reading on low SES kindergartens' emergent literacy as a function of learning context. *Computers & Education,* 51, 354-367.

Slangen, L, Keulen, H. V., & Gravemeijer, K. (2010). What pupils can learn from working with robotic direct manipulation environments. *International Journal of Technology and Design Education,* 21(4), 449-469. doi:10.1007/s10798-010-9130-8

Smith, D. (22 Sept. 2015). The 'grand experiment' behind NYC schools' new computer science program. Retrieved from: http://www.edtechmagazine.com/k12/article/2015/09/grand-experiment-behind-nyc-schools-new-computer-science-program

Smith, C. E., Tabors, P. O. Nicholson, P., & Kurland, B. (1994). SHELL: Oral Language and early literacy skills in kindergarten and first grade children. *Journal of Research in Childhood Education,* 10, 37-48.

Stanovich, K. E. (1986). Matthew effects in reading some consequences of individual differences in the acquisition of literacy. *Reading Research Quarterly,* 21, 360-406.

State of the States: state level policies supporting equitable K-12 Computer science education. (2017).

Stern, P. N. (n.d.). On solid ground: Essential properties for growing grounded theory. *The Sage Handbook of Grounded Theory,* 114-126. doi:10.4135/9781848607941.n5

Strauss, A. L. (1987). *Qualitative Analysis for Social Scientists.* New York: Cambridge University Press.

Strawhacker, A., & Bers, M. U. (2014). "I want my robot to look for food": Comparing kindergartners' programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education,* 25(3), 293-319. doi:10.1007/s10798=-014-9287-7

Studyvin, D., & Moninger, M. (1986, July). *Logo as an enhancement to critical thinking.* Paper presented at the meeting of the Logo 86 Conference, Cambridge, MA.

Sullivan, A., & Bers, M. U. (2015). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education,* 26(1), 3-20. doi:10.1007/s10798-015-9304-5

Sullivan, A., & Bers, M. U. (2016). Girls, boys, and bots: Gender differences in young children's performance on robotics and programming tasks. *Journal of Information Technology Education: Innovation in Practice,* 15, 145-165. doi:10.28945/3547

Sullivan, A., Kazakoff, E. R., & Bers, M. U. (2013). The wheels on the bot go round and round: Robotics curriculum in pre-kindergarten. *Journal of Information*

Technology Education: Innovations in Practice, 12, 203-219. doi:10.28945/1887

Sutherland-Smith, W. (2002). Weaving the literacy web. *The Reading Teacher,* 55(7), 662-669.

Sylvester, R., & Greenidge, W. (2009). Digital storytelling: Extending the potential for struggling writers. *The Reading Teacher*, 63(4), 104-113.

Sward, A. K. (2012). Re-learning in reading and writing – is it possible? *Procedia – Social and Behavioral Sciences*, 69, 104-113.

Tabors, P. O., Nicholson, P. A., & Kurland, B. F. SHELL: Oral language and early literacy skills in kindergarten and first-grade children. *Journal of Research in Childhood Education,* 10(1), 37-48.

Texas Administrative Code (TAC). (2011). Texas Essential Knowledge and Skills for Technology Applications, Title 19, Part II, Chapter 126.

Thoman, E., & Jolls, T. (2004). Media Literacy-A national priority for a changing world. *American Behavioral Scientist,* 48(1), 18-29. doi:10.1177/0002764204267246

Tuomi, P., Multisilta, J., Saarikoski, P., & Suominen, J. (2018). Coding skills as a success factor for society. *Education Information Technology,* 23, 419-434. doi:10.1007/s10639-017-9611-4

Turner, K. H., & Katic, E. K. (2009) The influence of technological literacy on students' Writing. *Journal of Educational Computing Research,* 41(3), 253-270.

Urquhart, C. (2013). *Grounded Theory for qualitative research: A practical guide.* Los Angeles, CA: Sage Publications.

Valverde-Berrocoso, J., Fernandez-Sanchez, M. R., & Garrido-Arroy, M. C. (2015). El Pensamiento computacional y las nuevas ecologias del aprendizaje, RED, *Revista De Educacion a Distancia,* 46(3). Retrieved from http://um.us/ead/red/46

Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies,* 1(2), 42-64. doi:10.21623/1.1.2.4

Vee, A. (2017). *Coding literacy how computer programming is changing writing.* Cambridge, MA: The MIT Press.

Vermer, A. (2001). Breadth and depth of vocabulary in relation to L1/L2 acquisition and frequency of input. *Applied Psycholinguistics,* 22, 217-234.

Vygotsky, L. S. (1978). *Mind in Society: The development of higher psychological processes.* Cambridge, MA: Harvard University Press.

Waite, J. (2017). Pedagogy in teaching computer science in schools: A literature review. Retrieved from https://royalsociety.org/~/media/policy/projects/computing-education/literature-review-pedagogy-in-teaching.pdf

Wang, X. C., & Ching, C. C. (2003). Social construction of computer experience in a first grade classroom: Social processes and mediating artifacts. *Early Education and Development,* 14(3), 335-361.

Wang, X-M., & Hwang, G-J. (2017). A problem posing-based practicing strategy for facilitating students' computer programming skills in the team-based learning mode. *Educational Technology Research and Development,* 65, 1655-1671. https://doi.org/10.1007/s11423-017-9551-0

Webster, T. E., & Son, J. (2015). Doing what works: A Grounded Theory case study of technology use by teachers of English at a Korean University. *Computers &*

Education, 80, 84-94. doi:10.1016/j.compedu.2014.08.012

White, M. D., & Marsh, E. E. (2006). Qualitative Content Analysis: A flexible methodology. *Library Trends,* 55(1), 22-45. doi:10.353/lib.2006.0053

Wing, J. M., (2006). Computational thinking. *Communications of the ACM,* 49(3), 33-35.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *Tech Trends,* 60, 565-568.

Yi-Ming Kao, G., Tsai, C.-C., Liu, C.-Y., & Yang, C.-H. (2016). The effects of high/low interactive electronic storybooks on elementary school students' reading motivation, story comprehension and chromatics concepts. *Computers & Education,* 100, 56-70.

Zhou, N. & Yadav, A. (2017). Effects of multimedia story reading and questioning on preschoolers' vocabulary learning, story comprehension and reading engagement. *Educational Technology Research and Development,* 65, 1523-1545. https://doi.org/10.1007/s11423-017-9533-2

APPENDIX A

Table 3
Scholarly Journals

Journal title	Total no. of articles	Total no. of themes	Key word and frequency				
			Literacy	Reading	Writing	Coding/ programming	Robotics
Tech Trends	1,143	23	5	10	6	0	2
Technology Knowledge and Learning	195	6	0	0	2	1	3
Computer Science Education	171	18	0	0	0	12	6
Themes in Science & Technology Education	93	7	0	1	1	0	5
Journal on Research and Technology in Education	209	8	1	1	0	1	5
Journal of Educational Computing Research	452	20	3	8	4	4	1
Educational Technology Research & Development	598	19	8	6	4	1	0
International Journal of Computer Science Education in Schools	25	8	0	0	0	8	0
Computers and Education	2,356	95	19	34	17	17	8

Totals	5,242	204	36	60	34	44	30

APPENDIX B

Table 4
Journal 1, Computers & Education

Article	Clusters of Meaning	Themes
Assessing elementary students' computational thinking in everyday reasoning and robotics programming	21	Humanoid robotics, Robotics for elementary students, computational thinking, designing systems, sequencing
Behavioral patterns of elementary students and teachers in one-to-one robotics instruction	31	Educational robotics, one-to-one robotics instruction, sequential analysis, communication skills, learning with robotics, computing, programming, problem solving, creativity, writing, informatics, engineering, subject and content integration
Computational thinking and tinkering: Exploration of an early childhood robotics curriculum	16	Early childhood robotics, programming, coding, cross curricular, early childhood literacy, computational thinking, problem solving, sequencing
Do ABC eBooks boost engagement and learning in preschoolers? An experimental study comparing eBooks with paper ABC and storybook controls	15	eBooks, reading fluency, reading comprehension, listening comprehension, engagement

Effects of computer assisted comprehension in second grade: A one-year follow-up study	24	Computer assisted text comprehension, sequencing, decoding, listening comprehension, reading comprehension, vocabulary, comprehension processes
Effects of digital dictionary Format on incidental acquisition of spelling knowledge and cognitive load during second language learning: Click-on vs. key-in dictionaries	12	Digital literacy, spelling, dictionaries, reading comprehension
Hypertext annotation: Effects of presentation formats and learner proficiency on reading comprehension and vocabulary learning in foreign language	14	Hypertext, reading comprehension, proficiency, vocabulary acquisition and comprehension, online reading, reading comprehension, reading process, electronic annotations, authentic texts
Improving literacy skills through learning reading by writing: The iWar method presented and tested	12	Literacy education, website, comprehension, enhanced literacy skills, interpret information, locate information on the web, information society, computers, reading, writing, speech technology, writing strategies, linguistic support, grammar
Moved to learn: The effects of inactivity in a Kinect-based literacy game for beginning readers	8	Literacy, on line books, reading skills, writing skills, language learning, digital reading, joint reading, sequencing

Positive technological and negative pre-test-score effects in a four-year assessment of low socioeconomic status K-8 student learning in computer-based math and language arts courses	15	Language, reading, writing development, test performance, computer-based work, sequencing, decoding, motivation
Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study	28	Programming, higher order thinking, problem solving, communicating, related subject areas and programming, social skills, communication skills, decoding
Promoting reading comprehension with the use of technology	32	Literacy, reading comprehension, fluency, phonics, vocabulary, reading practices, decoding, sequencing, reading comprehension
The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism	15	Programming, cognitive load theory, constructivism, cognitivist, learning theory, program visualization
The effect of digital storytelling on visual memory and writing skills	19	Comprehension, writing, reading, literacy, memory capacity, composition, writing skills, narratives
The effect of reflective learning e-journals on reading comprehension and communication in language learning	23	Reading comprehension, vocabulary, listening comprehension, communication skills

Understanding online reading through the eyes of first and second language readers: An exploratory study	9	Online reading, comprehension, decoding, sequencing, fluent reading, second language, meta-cognitive skills

APPENDIX C

Table 5

Journal 2, Educational Technology Research and Development

Articles	Clusters of Meaning	Themes
A computer-based spatial learning strategy approach that improves reading comprehension and writing	22	Reading comprehension, writing development, scaffolded practice, language arts curriculum, computer-based instruction, reading and writing skills, across the curriculum literacy
A problem posing-based practicing strategy for facilitating students' computer programming skills in the team-based learning mode	29	Programming, problem solving, collaborative learning, cognitive load, literacy, self-efficacy
Effects of multi-media story reading and questioning on preschoolers' vocabulary learning, story comprehension and reading engagement	36	Reading comprehension, storytelling, story reading, questioning strategies, vocabulary
Improving technology literacy does it open doors to traditional content?	59	Literacy, reading, programming, language arts, comprehension, sequencing, decoding
The influence of narrative and expository lesson text structures alternate measures of knowledge structure	32	Reading, writing, literacy, expository, writing, sequential occurrence, text structure, knowledge structures

APPENDIX D

Table 6

Journal 3, Journal of Educational Computing Research

Article	Clusters of Meaning	Themes
Assessing the impact of meta-cognitive training on students' understanding of introductory programming concepts	16	Programming, meta-cognitive, programming education, problem solving
Direct and indirect teaching: Using e-books for supporting vocabulary, word reading, and story comprehension for young children	41	E-books, reading, reading comprehension, vocabulary, decoding, sequencing, electronic storybooks, computerized dictionary, multimedia text, oral reading, oral discourses, written text
Does automated feedback help students learn to write?	12	Automated essay feedback, writing, writing lab, revision skills, computer learning skills, literacy
Evaluation of a motion-based platform for practicing phonological awareness of preschool children	39	Reading, phonics, decoding, comprehension, sequencing, visual discrimination

APPENDIX E

Table 7
Journal 4, Tech Trends

Article	Clusters of Meaning	Themes
Digital postcards from summer Camp	8	Writing, digital writing, computing, connected learning, literacy practices
Enhancing reading comprehension with student-centered iPad applications	23	Reading, reading comprehension, iPad, reading achievement scores, comprehension activities, literacy
Examining current beliefs, practices and barriers about technology integration: A case study	28	Language Arts, constructivist, technology, barriers for technology, 21st century skills, technology integration
Literature circles: A perfect match for online instruction	14	Reading, reading comprehension, vocabulary, sequencing, communication, presentations, literacy, literature online, reading fluency
Using nearpod in elementary guided reading groups	38	Literacy, reading, writing, language arts, digital natives, vocabulary, decoding, fluency, sequencing, text-to-speech, communication
Using technology to reach all learners	29	Reading, literacy, iPod, silent reading, reading fluency, literature, sequencing, decoding, phonics

APPENDIX F

Table 8
Data Source 2, Curriculum Standards

Standards	Clusters of Meaning	Themes
Code.org	12	Sequencing, decoding symbols and letters, understanding the order of letters and symbols to make words or lines of coding, communicating ideas through codes and symbols, predicting, encoding letters and creating a story
Computer Science Teachers Association	7	Computing systems, networks and the internet, data and analysis, algorithms and programming, impacts of computing
International Society for Technology in Education	7	Empowered learner, digital citizen, knowledge constructor, innovative designer, computational thinker, creative communicator, global collaborator
K-12 Computer Science Framework	9	Computing systems, networks and the internet, data and analysis, algorithms and programming, and impacts of coding
The Common Core Standards	12	Coincide with Language Arts standards; create, edit, and format text; evaluate, decompose, code, program, communicate ideas; develop and evaluate multimedia presentations, proper citations, grammar, electronically exchange ideas

APPENDIX G

Table 9

Data Source 3, Websites

Websites	Clusters of Meaning	Themes
2simple to code	57	Coders, multimedia stories, painting, compromising and designing code, publishing, reading, writing, sequencing, decoding, predicted
Code Academy	46	Coding, programming, web development, programming languages, skills of application, transfer, adapting, data science, sequencing, inferencing
Code.org	91	Pre-readers, debugging, coding, inferencing, cognitive skills, sequencing, decoding commands
Csunplugged	23	Computer science, intellectually stimulating, discipline, computing, binary numbers, decoding, sequencing, debugging

VITA

Renee O'Neal

EDUCATION & CERTIFICATION

Pursuing Doctorate in Literacy, Sam Houston State University graduation date December 2019; August 2017 ABD (All but dissertation).

Texas Superintendent Certification; February 2019.

Master of Education Degree/ Mid-Management Certification, December 1993 Stephen F. Austin State University, Nacogdoches, Texas.

Bachelor of Arts Degree in Elementary Education, Double Major Elementary Education and Special Education, November 1991, Houston Baptist University, Houston, Texas.

PROFESSIONAL EXPERIENCE

July 2017- Present- Principal for the Sam Houston State University Charter School System, 1908 Bobby K Marks, Box 2119, Huntsville, Texas 77341.
Responsibilities include overseeing the daily operations of four elementary campuses grades kindergarten-sixth grade, complete oversight of curriculum and instruction for all grade levels, coordination of all special programs, parent involvement activities and presentations, development of student and parent handbooks and recruitment and hiring of all teachers.

July 2005- June 2016 Principal, San Jacinto Elementary Conroe Independent School District Conroe, TX.

June 2001-June 2005 Assistant Principal, Ben Milam Elementary Conroe Independent School District Conroe, Texas.

June 2005 Elementary Summer School Principal Creighton Elementary Conroe, TX 77306.

June 2004 Elementary Summer School Principal Ben Milam Elementary Conroe, TX 77302.

July 1999-June 2001 Assistant Principal/Coordinator of Special Programs, PK-4

O.V. Calvert Elementary Aldine Independent School District Houston, TX

Dec.1991-May 1999 Teacher, Clifford Dunn Elementary School Aldine Independent School District Houston, Texas

PRESENTATIONS

O'Neal, R., Montenegro, M. (September 2015). Technology Applications and Second Language Learners. *International Conference of Educational Media*, Medellin, Colombia.

O'Neal, R. (November 2015). Hour of code and connection to literacy achievement. *Association for Educational Communications and Technology*, Indianapolis, Indiana.

O'Neal, R. (June 2016) Hour of Code and the connection to Literacy Achievement. JTel International Doctoral Fellowship summer school, Tallin, Estonia.

O'Neal, R., Wilson, T. (August 2016). Teachers' perceptions of the hour of code. *Education Technology World Conference*, Bali, Indonesia.

O'Neal, R. (November 2016). Teachers' perceptions of the hour of code. *National Council of Teachers of English*, Atlanta, Georgia.

O'Neal, R. (July 2017-present). All staff development for the Sam Houston State University Charter school staff from all four campuses.

O'Neal, R. (August 2018). Proposal for Doctorate of Literacy at Sam Houston State University. Title: Learning how to computer code/program and do the skills intersect with the developmental steps of learning to read and write?