

TEACHER SENSEMAKING: COMPUTER SCIENCE INTEGRATION IN THE
ELEMENTARY GENERAL EDUCATION CURRICULUM

by

Annmaria Tierno

A Dissertation Proposal Submitted in Partial Fulfillment
Of the Requirements of the Degree of
Doctor of Education

Middle Tennessee State University

November 2025

Dissertation Committee:

Dr. Lando Carter, Chair

Dr. Kevin Krahenbuhl

Dr. Brit Sikes

ACKNOWLEDGEMENTS

This dissertation represents the culmination of a journey made possible through the encouragement, wisdom, and generosity of many people. I would first like to extend my deepest appreciation to my chair, Dr. Carter, whose expertise, thoughtful feedback, and steady guidance continued to challenge me to think critically and refine my work. I am sincerely grateful to Dr. Krahenbuhl and Dr. Sikes for serving on my committee and offering valuable perspectives that strengthened this study.

To the faculty and my cohort at Middle Tennessee State University, thank you for fostering a supportive and intellectually engaging community. Your collaboration, laughter, and insight made this process both meaningful and rewarding. Each of you contributed to this work in ways that shaped not only my dissertation, but also my growth as a scholar and educator.

To my family, friends, and colleagues, your unwavering encouragement sustained me through the long and challenging moments. Scott, you have been my rock - your love, encouragement, and support kept me motivated even when I wasn't. Emma and Mary, your understanding and flexibility allowed me to balance professional, academic, and personal responsibilities throughout this process. Your support has been instrumental in helping me reach this milestone. I would also like to acknowledge my brothers Anthony and Stratos, whose good humor and spirited debates prepared me well for academic life by teaching me to argue a point – and to always have evidence to support it.

DEDICATION

This dissertation is dedicated to my mother, Kathleen Rose Carpenter. She was raised in a time when women had far fewer opportunities than we do today, yet she believed wholeheartedly that education was the key to independence, freedom, and a better society. From the time I was a small child, her advice to me was, “Annmaria, don’t marry a doctor, be a doctor.” My only regret is that I did not pursue this sooner or write fast enough for her to see it finished. She passed away shortly after I defended my proposal. My cheerleader has become my angel and her belief in the power of education continues to shape my path.

ABSTRACT

This basic qualitative study examined elementary general education teachers' perceptions and experiences related to integrating computer science into their curriculum. It explored the factors shaping those perceptions and analyzed how teachers' attitudes and experiences contributed to their sensemaking.

Three specific questions guided this inquiry:

1. How do elementary teachers understand and experience computer science integration into their general education curriculum?
2. What factors influence teachers' perceptions of their preparedness for integrating computer science into the general elementary education curriculum?
3. How do teachers perceive their preparedness for integrating computer science into the general elementary education curriculum?

The results indicated that through the iterative cycle of sensemaking, teachers interpreted the initiative through the lens of their prior experiences, refined their understanding through social interaction, and gradually internalized new learning that strengthened their self-efficacy. Findings also showed that teachers entered the process from different starting points based on their experience levels, but as they engaged with new information, they refined their practice and developed a stronger sense of competence in implementing the change.

This study highlights the importance of designing professional learning that acknowledges teachers' varied entry points, provides sufficient time for implementation, and strengthens their confidence. Future implications for sensemaking suggest the need for intentional supports that reduce uncertainty, help teachers interpret new initiatives, and build confidence as they refine their practice.

TABLE OF CONTENTS

LIST OF FIGURES	vii
LIST OF TABLES	viii
CHAPTER I: INTRODUCTION	1
Overview	1
Study Context	2
Statement of the Problem	3
Purpose of the Study	5
Significance of the Study	6
Theoretical Framework	5
Research Questions	6
Summary	8
CHAPTER II: REVIEW OF LITERATURE	10
Introduction	10
What is Computer Science	10
Computer Science Standards	23
Change Initiatives in K-12 Education	29
Sensemaking	37
Summary	46
CHAPTER III: METHODOLOGY	48
Research Purpose and Questions	48
Theoretical Framework	48
Subjectivity/Positionality Statement	50

Viewpoint of the Study.....	53
Research Site	55
Data Collection.....	57
Data Analysis.....	60
Trustworthiness and Rigor	61
Conclusion.....	65
CHAPTER IV: FINDINGS	67
Introduction	67
Data Collection.....	67
Participant Profiles	68
Reflexive Journal Notes	74
Interview Data Analysis	75
Research Question 1	78
Understanding Begins with Core Concepts.....	78
Instructional Guidance.....	83
Practical Realities of Computer Science Integration.....	86
Summary of Research Question 1	91
Research Question 2	91
External & Internal Factors Shape Teachers' Preparedness.....	92
Summary of Research Question 2	102
Research Question 3	102
Building Self-Efficacy.....	103
Summary of Research Question 3	112

Summary of Findings	112
CHAPTER V: DISCUSSION AND CONCLUSIONS.....	114
Introduction	114
Summary of Findings	115
Discussion of Findings	116
Research Question 1	116
Research Question 2	120
Research Question 3	123
Synthesis Across Research Questions	124
Connections Between Emerging Themes and Existing Literature.....	125
Recommendations for Practice.....	127
Recommendations for Policy	132
Recommendations for Future Research.....	132
Limitations of the Study	135
Conclusion	135
REFERENCES	137
Appendix A	155
Appendix B.....	158
Appendix C.....	159

LIST OF FIGURES

Figure 1. Theoretical Framework	49
Figure 2. Interpret → Refine → Internalize with Theories	126

LIST OF TABLES

Table 1. Participants	68
Table 2. Principal Coding Process	77
Table 3. Computer Science Definitions by Non-Computer Science Teachers	79
Table 4. Computer Science Definitions by Computer Science Teachers	79
Table 5. Knowledge and Understanding of Policy	84
Table 6. Relevance of Training	86
Table 7. Challenges Encountered with Integration	90
Table 8. Formal Networking Attitudes	94
Table 9. Messages Received - Support	95
Table 10. Building Self-Efficacy Sources	103
Table 11. Preparedness Rating Results	111

CHAPTER I: INTRODUCTION

Overview

Computer science education has become increasingly important over the course of the last decade (Code.org, CSTA, ECEP Alliance, 2022). While the concept of including it into the curriculum began in the 1980s with Seymour Papert’s work at the Massachusetts Institute of Technology, the push for its integration into the curriculum has only now just begun to take hold (Grove & Pea, 2018). In 2018, the state of Tennessee adopted computer science standards for the first time for elementary education (TN DOE, 2018). In 2022, the state of Tennessee revamped those standards and beginning with the 2024 – 2025 school year they require computer science integration into the general elementary curriculum (TN DOE, 2022). To do this effectively, schools will need to understand the changes and allow teachers time to make sense of the new content and how it fits with their current curriculum.

Darling-Hammond (2005) identifies change as a constant in schools and Bryk et al. (2010) notes that schools are complex institutions comprised of multiple interdependent subsystems. Policymakers must acknowledge that implementing a change initiative may disrupt those subsystems and that successful implementation efforts necessitate preparation, time, and knowledge of local needs (Fullan & Mundial, 1989). According to Chenoweth (2022) that “Schools improve as the result of people working together cooperatively over extended periods of time” (p. 45). Implementing change may also hinge upon teachers working collaboratively with one another to foster understanding and influence collective teacher buy-in. Empowering educators with knowledge and tools ensure that new subjects are effectively implemented in classrooms (El-Hamamsy et al., 2020; Oldham, 2021).

When implementing reforms, teachers face substantial intellectual demands, especially when the reforms are mandated (Fink & Stoll, 2005; Hargreaves, 2005). Thus, sensemaking, an iterative process of interpreting and constructing meaning from ambiguous or unclear situations, like those created by mandated reforms, is a key facet of implementing an initiative (Klein et al., 2007; Mills et al., 2010). In schools, sensemaking is triggered by change and impacted by social interactions, communication, professional growth, and past knowledge (Spillane, 2000; Coburn, 2005; Weick et al., 2005; Luttenberg et al., 2013). Teachers build their understanding of a change initiative by acquiring new subject knowledge, integrating new policies into existing frameworks, and refining their understanding through interactions with coworkers, external actors, and school administrators (Coburn, 2005; Pirolli, 2011; Soini et al., 2017). Engaging in the sensemaking process helps teachers make the abstract more concrete through interpretation of new stimuli through personal and social input (Dervin et al., 2008; Mills et al., 2010; Pirolli, 2011; Acona, 2012).

How educators make sense of change has a significant influence on how it is implemented (Stollman et al., 2020). As elementary general education teachers begin to integrate computer science into the curriculum, there will be many factors that influence their sensemaking and their capacity for integrating computer science into the general education curriculum.

Study Context

The goal of this study is to identify how teachers understand and perceive their preparedness for integrating computer science into the elementary general education curriculum and the factors that influence these perceptions.

The research will take place in the Spring semester of the 2024-2025 school year at a large, urban school system in Tennessee. The system serves over 30,000 elementary school students throughout the district and is made up of more than 70 elementary schools being taught by 2,800 teachers. Until the state mandated the inclusion of computer science into the elementary curriculum, the decision to offer the subject was determined by each school principal as part of their related arts rotation. Only 30% of elementary schools in the district included a dedicated computer science teacher in that rotation. After the state legislature mandated computer science for the elementary curriculum, the system required all elementary schools to create a team of *Integration Engineers* to attend a one-day train-the-trainer professional development workshop called Policy to Practice Training (P2P), to serve their individual schools as *Integration Engineers* to focus on integrating computer science into the general education curriculum. The *Integration Engineers* were charged with offering training throughout the 2023-2024 school year to prepare teachers for integration starting in Fall 2024.

Statement of the Problem

Significant progress has been achieved in incorporating computer science instruction into K-12 public education since Seymour Papert first introduced the idea in the 1980s (Grover & Pea, 2018; Code.org et al., 2022). Thanks to the advocacy efforts of numerous computer science organizations and Silicon Valley, there is growing recognition of the importance of implementing computer science at all K-12 levels (Code.org, 2019; Code.org, CSTA, ECEP Alliance, 2022). Many state legislatures have passed requirements for computer science education beginning in elementary school. However, a shortage of computer science teachers presents a challenge (Mason & Rich, 2019). Like many states, Tennessee hoped that graduates with computer science degrees would be interested in the field of teaching; however teaching

salaries struggle to compete with high-paying computer science positions. The average salary for computing jobs in Tennessee is \$90,354 compared to \$58,000 for an average teaching salary (Code.org, 2024a; Wegner, 2024). It is easy to see why those with a computer science degree would pursue opportunities other than teaching. Therefore, the responsibility for integrating computer science into elementary education will primarily fall on in-service teachers who often lack content knowledge for this subject. To implement this effectively, K-12 teachers need to understand their capacity to undertake this change initiative.

Teachers may be accustomed to the constant change in education, but it is not without its difficulties (Darling-Hammond, 2005; Hargreaves et al., 2007). Implementing change gradually, encouraging teacher support and buy-in, and making sure teachers have access to ongoing professional development are not easily done and require additional resources and planning to do it with fidelity (Hall, 1992; Ertmer, 1999; Coburn, 2005; Darling-Hammond, 2005; El-Hamamsy et al., 2020). The difficulty of scaling the change is another factor. There is often a tense and unpredictable relationship between policy and practice that further complicates bringing the change to scale (Elmore, 2016). Local issues also impact implementation, which can vary from class to class, making large-scale change difficult because there is no one-size-fits-all solution (Fullan & Mundial, 1989; Hall, 1992). Implementation includes policymakers depending on other interested parties and independent trainers to interpret the policy messaging, which can weaken or dilute the change's objective and leave each school with a distinct interpretation that may or may not be consistent with the policy's intent (Coburn, 2005). There is also the belief that there are too many standards and topics to be covered in school and adding computer science is not an essential for elementary school (Marzano, 2003). Challenges exist with teacher buy-in as well. In particular, teachers need to be committed and motivated to implement the changes

and believe there is merit to doing so (Haney et al., 2002; Lee & Min, 2017; Macy & Wheeler, 2021; Grebing et al., 2023). This is a big hurdle, particularly because their sense of self-efficacy regarding computer science is important to their motivation to implement the change (Lee & Min, 2017). Finally, teachers will navigate their way through the sensemaking process, coming to an understanding about the change initiative. Besides the initiative itself, external factors will influence teachers' thinking along with their pre-existing knowledge and professional development experiences related to computer science (Pirolli, 2011; Ancona, 2012). The combination of these elements will shape teachers' sensemaking about their capacity for integrating computer science into the general education curriculum.

Purpose of Study

Implementing change in education can be difficult for many reasons. Research on change initiatives has revealed that the success of new initiatives depends on the ability of educators to gain knowledge, try new things, and adapt concepts to their classrooms (Darling-Hammond, 2005). Curriculum changes necessitate fresh learning for the teachers who will carry them out (Pietarinen et al., 2019). However, there are limited opportunities for teachers to learn more about the content they will implement (Cohen & Ball, 1999). Sensemaking begins when people feel conflicted about the execution of a new change (Lowell et al., 2024). Teachers draw on personal and social resources in the process of sensemaking (Allen, 2023). According to Siciliano et al. (2017), the process by which educators interpret reforms and their ability to implement it is not given enough consideration.

Therefore, the purpose of this qualitative study is to explore the perceptions and experiences of elementary general education teachers regarding their capacity to integrate computer science into their curriculum. The study aims to investigate how teachers make sense

of a new curricular initiative – in this case, computer science integration – what factors influence that sensemaking, and how sensemaking impacts their assessment of their own preparedness to carry out a change initiative. The findings will offer potential insights to support the integration and implementation of computer science curriculum in primary grades as well as implementation and sustainment of other elementary teacher preparation and professional development endeavors.

Significance of the Study

The effectiveness of change initiatives depends upon how well teachers understand the purpose of the change, commit to making it, and develop the skills necessary for successful implementation (Fullan, 1989). This requires teachers to tie the concrete and abstract together through the sensemaking process (Weick et al., 2005). Teachers do this through an iterative process that includes their internal understanding of the change based upon their prior knowledge and professional development experiences and their external understanding that comes from the policy, standards, messages, and guidance they receive (Spillane et al., 2002). Teachers are often intimidated by computer science, and there is a lack of information available to guide elementary school teacher preparation for integration into the general education curriculum (El-Hamamsy et al., 2020; Hutchison et al., 2021; Mason & Rich, 2021; Oldham, 2021). Therefore, more research is required to understand how teachers interpret and make sense of reforms and their capacity to implement them.

This study examines how teachers prepare for and understand the integration of computer science into the general education curriculum, perceptions of their preparedness for implementation, and the factors that influence their perceptions. The open-ended interviews will provide insight into how teachers make sense of a reform initiative for a subject matter with

which they are not familiar. Implications of this study may be adding to current research about the kinds of activities and supports that work best for teachers learning to integrate a new subject matter.

Theoretical Framework

The theoretical framework for this study is grounded in sensemaking theory, which explores the interaction between action and interpretation (Weick et al., 2005). Like Spillane et al. (2002), I will focus on the study of sensemaking in regard to reform initiatives. According to Weick (1995), people need to identify a relationship between two different facts or experiences to understand a change and give it meaning; without which neither element has any significance. Spillane et al. (2002), outline a framework that acknowledges three key elements in implementation: 1) individual and social context, 2) situational influences, and 3) policy influences. These elements will guide my inquiry as I investigate the relationships between teachers knowledge and external influences as they relate to new change policy to be implemented.

Research Questions

This study will consider the introduction of a change initiative into the curriculum and explore individual teacher's understanding of the policy regarding integration. I will investigate the prior knowledge and professional learning teachers had at the time they encountered the change. Finally, I will consider the impact of the external influences that supported and guided teachers in their sensemaking toward their capacity to implement the integration of computer science into their elementary, general education curriculum. The following research questions serve as the drivers of this study:

RQ1: How do elementary teachers understand and experience computer science integration into their general education curriculum?

RQ2: What factors influence teachers' perceptions of their preparedness for integrating computer science into the general elementary education curriculum?

RQ3: How do teachers perceive their preparedness for integrating computer science into the general elementary education curriculum?

Summary

The society in which we live is fast-paced and constantly changing. Today's educators are often asked to adapt and make changes to their curriculum (Darling-Hammond, 2005). One such change that has taken root in education priorities in many states, including Tennessee, is the inclusion of computer science in the curriculum. However, this change is complex and not without its challenges. Preparing to implement these changes will necessitate the need for careful planning, collaboration, and examination of the needs of each school community (Fullan & Mundial, 1989). Among those needs are qualified teachers for computer science. With teacher shortages around the country, finding those qualified teachers is much harder and calls for more feasible solutions for implementation (Mason & Rich, 2021; Vegas et al., 2021). At the elementary level, it will be the responsibility of general education teachers to integrate computer science into their regular lesson plans (El-Hamamsy et al., 2020; Oldham, 2021). To do so effectively, teachers will have to make sense of the policy change and acquire new knowledge.

While the process is often overlooked, it is important that we understand how teachers make sense of change initiatives that they are expected to implement within their classrooms (Siciliano, 2017). This study will examine the ways in which teachers at a large, urban school system make sense of a change initiative and the factors that influence their understanding. It

relies on a theoretical framework in sensemaking that examines internal and external influences that impact understanding a change policy. Addressing the complexity of implementing change by shining a light on teacher sensemaking will provide insight into the support and training necessary to ensure teacher preparedness in the future.

CHAPTER II: REVIEW OF LITERATURE

Introduction

Significant advances have been made to integrate computer science education into K-12 public education (Code.org et al., 2022). These changes will require that in-service teachers have a better understanding of computer science which means professional development initiatives should equip teachers with the skills and confidence needed to do this effectively (Dorotea et al., 2021). This review of the literature will discuss what computer science is, implementation of change initiatives, and how sensemaking takes place in education.

What is Computer Science?

It is critical to understand what computer science is and is not to successfully integrate it into the curriculum. Early definitions are too broad, simplifying computer science to "...the study of computers and all the phenomena associated with them..." (Grover, 2020, p. xiv). The definition of computer science evolved to be synonymous with programming, or coding, as a result of Code.org's early advocacy for its inclusion in all subject areas (Oldham, 2021). Rapaport (2017) reviewed multiple definitions for computer science which ranged from it being the study of computation, to "...a natural science that studies procedures" (p.73), to the study of computational thinking, and several other iterations in between; ultimately concluding that there is no one way to define computer science. According to Oldham (2021), there is more consensus on what computer science is not than what it is.

It is generally acknowledged that fundamental computer skills like typing, word processing, presentation software, and spreadsheets are not the same as computer science (Fluck et al., 2016; Oldham, 2021; Wohl et al., 2017). These are the kinds of skills that are typically taught as part of the information and communication technologies (ICT) curriculum, commonly

known as computing but differs significantly from computer science (Fluck et al., 2016). Computer science is its own distinct subject made up of related areas (Tucker, 2003). Decisions about computer science curriculum are determined by the states based upon their needs and will often overlap with information technology, math, computational thinking, science, literacy, and digital citizenship (Oldham, 2021). Understanding computing-related ideas improves the understanding of information systems, how technology functions, and problem-solving techniques (Dorotea et al., 2021). One of the ultimate goals of computer science is to move students from being consumers of technology to being creators of it (Funke et al., 2016; Oldham, 2021). To do so, a steering committee composed of computer science professionals, states, and school districts created a framework that local governments can use as a guide when creating computer science standards that align with their prioritized needs. The *K-12 Computer Science Framework* (2016) contains five essential elements: computing systems, networks and the internet, data and analysis, algorithms and programming, and impacts of computing (Alano et al., 2016; Mason & Rich, 2019; Smith & Workman Publishing, 2020). When considering what computer science is not, as well as the overlapping objectives and curricula, agreement can emerge around the following definition of computer science: “Computer science refers to the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society” (Wilson et al., 2010, p. 10). Knowing what computer science encompasses is crucial to determining what knowledge teachers need to possess in order to teach it.

Computer Science in K-12 Education

As early as 1971, the use of computers as teaching tools was taking place in high schools around the country. Students were developing programs that simulated moon landings, enhanced

daily school record keeping, and monitoring athletic teams to improve their performance against opponents (UPI, 1971). In 1978, Apple partnered with the Minnesota Education Computing Consortium to place 5,000 computers in schools throughout the state, and launched Apple on its path toward the goal of installing their computers in every elementary and high school in the country (Buck, 2017). Initially, computers were largely seen as a tool for instruction (UPI, 1971). However, Steve Jobs believed that every student should have access to computers and Apple started the *Kids Can't Wait* program which managed to get computers into every Californian school with more than 100 students (Buck, 2017).

About the same time that Jobs and Apple computer were expanding their reach into schools, Seymour Papert was creating Logo, one of the earliest programming languages designed for children (Bers, 2008). At its foundation Logo was more than a language, it also provided a powerful environment to think about learning (Solomon, 2020). In his seminal work, *Mindstorms* (1980), Papert articulated two primary themes, “that children can learn to use computers in a masterful way, and that learning to use computers can change the way they learn everything else” (Papert, 1993, p. 8). This is a first glimpse of the multidisciplinary nature of computer science. Prior to this writing, Papert had studied alongside Piaget for five years and developed an alternative theory of constructivism that he called constructionism (Stager, 2016).

Constructionism is both a learning theory and educational strategy that aligns with early childhood education’s use of manipulatives to engage learning (Bers, 2008). Papert partnered with LEGO to make the first robotics system for children to program, which they named Mindstorms (Stager, 2016). Papert emphasized learning in which children created their own knowledge about the outside world and by using their bodies and other objects available to them (Ames, 2018). Papert’s work ignited the *One Laptop per Child* initiative that sought to provide

all children, from Maine to Australia, with access to laptop computers which contributed to his being called the Father of Educational Computing (Stager, 2016).

With *Adventures in Supercomputing*, a K-12 program designed to encourage underrepresented students and their teachers toward science, math, and computing, a greater emphasis was placed on computing education in the late 20th century (Oliver et al., 1993). These efforts, through the Department of Energy, were directed at school reform that included integrating more technology and science into the curriculum in an attempt to address the future success of those industries in the United States. The President's Committee of Advisors on Science and Technology (PACST) formed the Panel on Educational Technology to advise the President on strengthening K-12 education through the application of technology (Shaw, et al., 1998). The Panel concluded that increasing funding for educational technology research and equipment deployment to the nation's schools should be a priority. Also, during this time period, the internet became available to the general public and K-12 schools. Microsoft provided a variety of resources to schools through their *Microsoft Focus on K-12* initiative that included lesson plans, activities, and additional information for integrating technology into the classroom (PR Newswire, 1995). Schools saw a considerable rise in the frequency of computer use in the classroom. Student-to-computer ratios decreased significantly, with the exception of Title 1 schools, where the ratio was three times greater than non-Title 1 schools (Swain & Pearson, 2003).

Moving into the 21st century, a report was released by the Department of Commerce finding that people with lower incomes, less education, the elderly, and people living in both urban and rural areas had less access to the internet (Lentz, 2000). There was a great deal of concern about the digital divide and its impact on student achievement. Historically a slow-changing industry, education saw significant transformation, in the early part of the new century,

to keep up with the nation's general increase in internet use (Huang & Russell 2006). Disparities persisted between socioeconomic groups in relation to internet access and computer access with those students from lower incomes having less access and older equipment than their wealthier counterparts (Swain & Pearson, 2003). These disparities still exist today, but they are shrinking (Schiel, 2024).

A full thirty years after Steve Jobs failed at lobbying Congress to change the tax laws so that Apple could donate computers to all 100,000 schools in the country, high tech business leaders banded together in a reinvention of Jobs efforts to bring this change to scale (Buck, 2017; Morel, 2019). In 2013, Code.org joined with other tech leaders to bring computer science to every public school in the United States (Wilson, 2013). They made equity and access one of the core values of their work (Code.org, 2023). Lending a powerful endorsement to Code.org and its backers was then-President Obama who declared computer science to be just as important for all K-12 students as reading, writing, and math (Oldham, 2021). In 2016, in order to increase the number of students receiving computer science education in grades K-12, he started the *Computer Science for All* initiative (Lodi, 2020). Today, Code.org has developed a broad coalition for computer science advocacy and has worked with legislators across the country to adopt policies that support computer science education (Code.org, 2022).

There is growing recognition that computer science education is a crucial component of K-12 education. In a Google/Gallup poll (2016) half of all teachers and two-thirds of parent responders identified computer science as important as other core subjects and anticipate its inclusion into the curriculum will continue to grow in the future (Ozturk, et al., 2018). Over the last 50 years, the United States has seen the birth of computer science as a discipline. From early

efforts using computer technology in high schools to the adoption of state standards for our youngest learners.

Detractors argue that schools should concentrate primarily on implementing the most essential elements of education (Schmoker, 2018). Concerns have been raised that the focus on skills has diminished foundational knowledge and the elementary school curriculum, which ought to emphasize the strong connection between knowledge and critical thinking (Christodoulou, 2014; Hirsch, 2017; Schmoker, 2018). They articulate how these practices widen the achievement gap for disadvantaged students (Hirsch, 2017; Schmoker 2018). Additionally, they acknowledge that technology has made a significant impact on the world, changing the way that we live but that we are educating children for a world that we cannot yet imagine (Wiliam, 2018).

A balanced approach to education can incorporate both traditional skills and the growing importance of computer science, while also addressing the critical issue of diversity (Franklin, 2020). The K-12 Computer Science Framework (2016) was designed to lend itself to a wide range of implementation options and computer science has a multi-disciplinary foundation that can enhance critical thinking (Berry, 2013; Ronan & Williams, 2024). Schmoker (2018) admits that it is also necessary that we prepare students for college, career, and citizenship as part of their education. While valid observations have been made that one of our challenges lies in our limited ability to accurately predict the jobs that will be in demand in the future, we know that many current and future jobs will employ the critical thinking and problem-solving skills inherent in computer science instruction and are deemed necessary by scholars (Christodoulou, 2014; Schmoker, 2018; Wiliam, 2018; Franklin, 2020; Ronan & Williams, 2024). Additionally, early computer science education tackles educational disparities and research shows that students

who study computer science do better in other subjects (Santo et al., 2019; Code.org, 2024).

Hirsch (2017) also reminds us that there is no one age where a topic is appropriate and that the state of their relevant knowledge is what determines when they should learn something. Children are already surrounded by the concepts and knowledge that are the foundation of computer science, so integrating it into the curriculum is an appropriate choice (Grover, 2020).

Computational Thinking

Much of the curriculum used to teach computer science is centered around the principles of computational thinking (Angeli et al., 2016; Bers et al., 2023; Falkner & Vivian, 2015; Fluck et al., 2016; Ozturk et al., 2018; Shine et al., 2021). Seymour Papert (1993) was the first to write about “computational thinking,” but he offered no definition (Lodi & Martini, 2021). In her seminal article, Jeannette Wing (2006), defines computer science as, “the study of computation” (p. 34) and explains computational thinking as “solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (p. 33). Bers (2012) summarizes computational thinking as, “the range of mental tools, found in the breadth of the field of computer science, that involve analytical thinking” (p. 69). The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) collaborated on a definition that describes computational thinking as a process used to solve problems which includes the ability to solve problems by using a computer to logically analyze and represent data, think algorithmically, identify and implement solutions, and transfer the problem-solving process to a wide variety of other problems (ISTE & CSTA, 2011). ISTE (2023) further identifies computational thinking as, four pillars essential for all students that include: decomposition, pattern recognition, abstraction, and algorithms. These four pillars are widely acknowledged as the key elements of computational thinking that form the of

computer science by academics and producers of computer science education resources, like Barefoot Computing and Code.org (ISTE, 2023; Ma et al., 2022; & Rana, et al., 2022).

According to Wing (2006) computational thinking is an essential skill that should be added to every child's analytical repertoire. It is a problem-solving skillset with universal applicability that allows children to think in logical ways to solve problems (Bers, 2022). By examining each of the four pillars, it is easy to understand why Wing (2006) believed that it needs to be a skillset embraced by all. The word algorithm has become used quite frequently when discussing social media and how one's news feed is populated. It is a term that people use to explain why they are not seeing what they would like to on social media or why they are not being promoted to others. While widely used, not everyone understands what an algorithm is. An algorithm is, quite simply, a set of rules or instructions that follow a specific sequence (Barefoot, n.d.). All computer coding is an algorithm. Children are used to algorithms, they practice them regularly in school from the order of their day to classroom protocols, to solving mathematical equations. Teachers use algorithmic concepts in lesson planning. One of the earliest mathematical concepts taught is patterns. According to Code.org (n.d.), a pattern is, "A theme that is repeated many times." Young children's books also include repeating patterns and identifying patterns is needed for data analysis and coding. Patterns are the building blocks of creating loops in coding, students need to understand patterns in order to understand loops. Teachers look for patterns in student data to determine which concepts require re-teaching and when to use scaffolding in a lesson. The last two concepts in computational thinking are abstraction and decomposition. Abstraction helps simplify problems by identifying the elements critical to the problem itself (ISTE, n.d.). Computer scientists use abstraction to identify necessary details and manage information (Barefoot, n.d.). Students use abstraction when they

create graphic organizers, identify themes in a story, and solve word problems. Teachers abstract from the standards when they design a rubric to specify the outcomes that they expect from student learning. Finally, decomposition is used to break a problem down into smaller, more manageable parts (Code.org, n.d.). The design process used by computer scientists, engineers, and project managers is an example of decomposition (Barefoot, n.d.). When studying money, students break down monetary values into smaller ones to determine which coins to utilize to get the desired amount. The aforementioned scaffolding is an example of decomposition that a teacher might use to make something more easily understandable for students.

Jeannette Wing (2006), stated that computational thinking is, “A way that humans, not computers think” (p. 35) and is the way in which we solve problems. While this way of thinking is a cornerstone of computer science, computational thinking is also used in multidisciplinary ways (Berry, 2013). While computational thinking is employed throughout the curriculum, that does not mean it is easily transferred and therefore does not require direct instruction. Lodi & Martini (2021) clarify that while computational thinking does have a transversal role with other subject areas, that it should not be sidelined to those areas. They offer the example of math, physics, and chemistry. Math is of critical importance to both subjects, but we do not leave the instruction of math solely in the context physics and chemistry, it has its own place in the curriculum. The same is true for computational thinking. Lodi and Martini (2021) specified that “CT must be understood in its proper context, which is that of CS [computer science]”.

Computer Science Framework

In today’s world, computers and related technologies are commonplace. Because computing is so ubiquitous, it has profoundly changed how most professions do their work and how many people live their personal lives. The global economy has changed due to computer,

necessitating the demand for a workforce with STEM-related skills and education (Oda et al., 2021). Over 500,000 computer jobs are available in the United States, spanning all industries, and growth is predicted to continue (Code.org, n.d.). Almost half of the 7 million Americans who work in fields unrelated to STEM use computers for their professions (K-12 Computer Science Framework, 2016). Tennessee presently has 5,779 open computing jobs with an average salary of \$90,354 (Code.org, 2023a). It is easy to see why the demand for a workforce skilled in computing is needed. However, computer science is not included in compulsory education in the United States as it is in many other countries around the world (Oda et al., 2021).

Global education systems are under increased pressure to incorporate computer science into their curricula due to the demands of the future workforce (Conde-Zhingre et al., 2023). At present, only 53% of all high schools in the United States teach computer science, despite having 93% of parents wanting it taught in school (Code.org, 2023a). Additionally, computer science is one of the most popular courses offered in school, ranking third after visual and performing arts (Code.org, n.d.). Students can use computer science as a tool for study and expression, becoming active members of society that are not just consumers of technology but also creators (K-12 Computer Science Framework, 2016; Yang et al., 2020; Conde-Zhingre et al., 2023). Curriculum reform efforts are needed for the inclusion of computer science beginning with the elementary level (Oda et al., 2021). Early involvement in computer science dispels misconceptions about who belongs in the field and enables students to become fluent across many years (K-12 Computer Science Framework, 2016).

Several organizations came together to form the K-12 Computer Science Framework Steering Committee to promote the incorporation of computer science into schools by laying the foundation for K-12 computer science courses (K-12 Computer Science Framework, 2016;

Conde-Zhingre et al., 2023). The framework provides guidelines to be used in developing standards and curriculum (Yang et al., 2020). Rather than representing abstract, broad ideas, the framework focuses on five core concepts that represent particular areas of disciplinary importance (Conde-Zhingre et al., 2023). The core concepts are:

1. Computing Systems
2. Networks and the Internet
3. Data and Analysis
4. Algorithms and Programming
5. Impacts of Computing

The framework also identifies several practices that students who are proficient in computer science take to fully interact with the fundamental ideas of the field (K-12 Computer Science Framework, 2016). The seven practices include:

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 core of computer science practices is computational thinking, which is outlined in practices 3-6 (K-12 Computer Science Framework, 2016).

The framework specifies the knowledge and skills that students must possess by the conclusion of the second, fifth, eighth, and twelfth grades (Conde-Zhingre et al., 2023). It is

intended to provide a starting point for the development of computer science education standards for K-12 students by all states, districts, and organizations (K-12 Computer Science Framework, 2016). Additionally, the framework offers guidance for developing standards in the local context. These competencies are regarded as essential for 21st century education because they equip students to take on the possibilities and challenges of both the present and the future (Conde-Zhingre et al., 2023).

Technology Integration

When one is aware of the history of computers in education, it is easy to comprehend why there is confusion between computer science and technology integration. Originally, the goal of Steve Jobs and others was to get computers into the hands of children, computer science was not considered separate from integration in K-12 schools at the time. Additionally, there was not an abundance of computers in the classrooms like there is today, so teaching with computers was not a priority. However, that began to change in the early 2000's with the No Child Left Behind (NCLB) Act of 2001 which added the Enhancing Teaching Through Technology (ETTT) initiative (Lowther et al., 2008). Billions of dollars has been invested by the United States government in school technology infrastructure (Liu et al., 2016).

In the modern world, providing technology to students is essential (Delgado, 2015). The NCLB program and its offshoot, the ETTT initiative, provided states with the resources to expand those efforts from elementary to high school (Ross, 2020). More students have access to computers in the country's classrooms thanks to the push for 1:1 devices, which is the ratio of digital devices to students (Delgado, 2015; Inan & Lowther, 2010). A 1:1 student-to-device ratio has been linked to improved teaching, higher student achievement, and greater student engagement (Liu et al., 2016; Inan & Lowther, 2010). Having access to computers has made it

easier to finish schoolwork, find material for projects and interact with peers and teachers in an effective way (Ross, 2020). In short, incorporating instructional technology into the classroom opens up new possibilities for achieving the desired learning goals (Consoli et al., 2023).

One of the most important areas of educational technology research in recent decades has been the integration of technology into the classroom (Consoli et al., 2023). The main motivation behind integrating technology into K-12 education is to prepare students for the workforce and enhance their knowledge and abilities (Lowther et al., 2008). Technology integration, also commonly referred to as information and communication technology (ICT), is the general application of technology that distinguishes itself from computer science (Fluck et al., 2016). It is also not uncommon to see ICT and computer science taught in tandem, particularly since both make use of computers. Additionally, technology integration has been defined in multiple ways but does have specific elements that are identifiable in each (Hew et al., 2007). Ultimately, technology integration is the process of using tools and applications embedded into instructional practices to support learning across the curriculum (Consoli et al., 2023; Fluck et al., 2016; Hew et al., 2007).

Coding

There is a widespread misconception that coding and computer science are synonymous (Wing, 2006; Oda, 2021). Grover (2020) explains that because coding involves both abstract and concrete elements, computer science and coding are “unique subjects” (p. xvii). Bers (2018), explains coding as a way to strengthen and engage computational thinking and vice versa, describing a more symbiotic relationship between the two. Coding is also referred to as a new type of literacy (Bers, 2018; Jacob & Warschauer, 2018; Woo & Falloon, 2023). However, the Computer Science Framework includes programming, which is a term often used

interchangeably with coding, as a core concept (K-12 Computer Science Framework, 2016). Since the framework is a standard bearer for most computer science study, it is reasonable to state that computer science is an umbrella term under which several elements, including coding, exist.

Computer Science Standards

Tennessee State Standards

As previously stated, there is growing recognition that computer science education is a critical component of K-12 education, but elementary education has been slow to launch its implementation into the curriculum (Ozturk et al., 2018). The K-12 Computer Science Framework Steering Committee (2016) created guidelines for teaching computer science intended to be used by policymakers at the state and/or district level, curriculum developers, teachers, and other supporting organizations. The framers purposely left it open for modifications to the local context.

In 2018, the state of Tennessee adopted the *Digital Readiness, K-8 Computer Science Standards*, in a first attempt to address the requirement that students acquire digital skills and lay the groundwork for future development. The standards focused on six major strands for comprehensive digital readiness to be included into the curriculum at each grade level. It was the first time the state prioritized an emphasis on computer science instruction in the elementary school classroom. They did not, however, create a pathway for an endorsement in computer science for educators at that level. In 2022, the state revisited the standards and amended them as *Computer Science, Tennessee K-12 Computer Science State Standards*, restructuring the standards specific to each grade level and reorganizing the original six strands into six core concepts while eliminating several standards at each elementary grade level. Additionally, they

have required that all schools begin integrating the standards into their core content curriculum beginning in the 2024-2025 school year. The intent for these standards is for all students to have access to computer science education and for computer science to become an essential component of K-12 education.

Alignment to General Curriculum

There are three general approaches to implementing computer science education: (a) independent subject, (b) integrated into existing subjects, (c) computer science based curriculum with a cross-curricular emphasis (Oda et al., 2021). Most operate inside a framework that establishes a conceptual basis for computer science (Webb et al., 2017; Yang et al., 2020; Kwon et al., 2021). Appropriately trained teachers are critical to successful curriculum integration but there is a shortage of qualified teachers available to teach computer science (Webb et al., 2017; Mason & Rich, 2021; Vegas et al., 2021). Integration will become part of the in-service teacher's already demanding role of delivering content, particularly in elementary education (El-Hamamsy et al., 2020; Oldham, 2021). This essentially leaves teaching computer science through existing subjects the best option for most elementary classrooms.

Due to the overloaded elementary school curriculum, teaching computer science by grade level teachers becomes increasingly challenging (Davis, 2019; Novak & Khan, 2022). Schmoker (2018) promotes a streamlined educational approach removing initiatives that do not place an emphasis on reading and writing standards. He discusses Conley's four habits of mind, which form the cornerstone of student learning across all subject areas:

1. Read to infer/interpret/draw conclusions.
2. Support arguments with evidence.
3. Resolve conflicting views encountered in source documents.

4. Solve complex problems with no obvious answer.

However, through integration, elementary school teachers can take advantage of the significant overlap between computational thinking and other subject areas (Waterman et al., 2020).

Computer science, with its core concepts stemming from computational thinking and its focus on problem-solving lends itself well to several content areas, including literacy goals. Additionally, incorporating computer science concepts into math, English language arts (ELA), science, and social studies is a strategy that can be used without adding to the general curriculum (Webb et al., 2017; Davis, 2019; Novak & Khan, 2022).

When implementing computer science into the existing curriculum, there is no need for a major overhaul of general education curriculum (Davis, 2019). For example, picture books can be used to support computer science concepts such as sequencing, which is a critical concept in reading comprehension. Computational thinking is a foundational component of computer science, it is essentially the process of problem solving which can be easily incorporated into science and math practice (Wing, 2006; Waterman, 2019). Pattern Recognition is an example that transcends most elementary school subject areas such as, math, literacy, science, and social studies, and can be mapped to the standards for each (Williams, 2021). Teachers can teach computer science and computational thinking ideas without adding a new subject to the elementary school curriculum by utilizing the computer science standards and coordinating them with the general education requirements. Teachers can engage students in problem solving, customize learning, and make connections between new information and previously learned concepts by incorporating computer science into the general education curriculum (Webb et al., 2017).

Computer Science Pedagogy

Pedagogy, is defined as the “...method and practice of teaching...” and can be simplified to “...the how and why of teaching...” (Muhammad & Hollie, 2012, p. 79). The primary challenge in relation to computer science, especially in K-12 education, is that there is no prescribed way in which computer science is taught (Northrup et al., 2022). However, it is advised that the proposed strategies be age appropriate and realistic for students (Friebronn-Yesharim & Ben-Ari, 2018). Therefore, determining the most effective and engaging strategies to include for students and integrating those strategies into the classroom is critical (Ryoo, 2019). Most learning theories include information on how to organize training to optimize learning for students which allows teachers to adapt their pedagogy to the content, regardless of their preferred learning theory (Ertmer & Newby, 2013). However, for integration into the general curriculum, a pedagogical shift may be needed to teach computer science (Northrup et al., 2022).

In order to effectively teach computer science concepts and practices, teachers need to give significant thought to the content and the delivery of their instruction (Kandemir et al., 2020). The Computer Science Framework includes five core concepts of disciplinary importance: computing systems, networks and the internet, data analysis, algorithms and programming, and impacts of computing (Conde-Zhingre et al. 2023). Computer science does not have one simple definition and is sometimes oversimplified as all things related to using computers (Grover, 2020; Rapaport, 2017). Teachers agree that computer science is a necessary component for K-12 curriculum because it permeates all aspects of life and work, although it is not fully embraced as necessary at the elementary level (Mertala, 2021; Northrup et al., 2022). Additionally, a large number of today’s elementary school teachers never received any computer education during their school years (Kim et al., 2022). Research on computer science instruction

in elementary schools is clearly needed and the success of such instruction is greatly dependent on the methodology and content knowledge of the teacher (Bjursten et al., 2022).

The best pedagogical practices must also be rooted in how students learn. Computational thinking, the foundation of computer science, is used for problem-solving (ISTE & CSTA, 2011). The entire process of problem solving requires critical thinking (Orhan, 2022).

Background knowledge is a prerequisite for critical thinking and without it we cannot develop sound reasoning skills (Willingham, 2021). Therefore, opportunities need to be provided to foster the development of background knowledge in students.

Teachers assist students in creating knowledge by utilizing both their existing and new experiences to connect them to learning that is relatable to their everyday lives (Ryoo, 2019; Alaijlan et al., 2023). As we learned from constructivism, creating meaning from experience is critical to learning (Ertmer & Newby, 2013). Papert, who is well-known in the field of computer science education, took constructivism one step further indicating that students construct knowledge through hands-on, interactive activities, a theory he called constructionism (Resnick, 2017). Willingham (2021) identifies exposure as a simple method for building knowledge, particularly when it is conceptual and includes extended practice that is spaced out over time. Taken together these ideas translate to pedagogy through teaching strategies selected for building knowledge. Two experiential practices that fit into the constructivist/constructionist approach to teaching, namely unplugged activities and physical computing, are popular techniques for increasing students' knowledge for computational thinking and computer science (Mertala, 2020; Huang & Looi, 2021; Northrup et al., 2022; & Alaijlan et al., 2023). Unplugged activities, a term created by the Computer Science Unplugged movement, are those activities which do not require a computer or other machine learning (Huang & Looi, 2021). In contrast to unplugged activities,

physical computing makes use of tangible objects – like robots – as manipulatives to create concrete conceptual representations (Mertala, 2020; Williams, 2021). These powerful pedagogical practices lend themselves well to the teaching and strategies used for developing computational thinking in computer science, including problem-based learning, scaffolding, cooperative/collaborative learning, and project-based learning, which are employed to cultivate the skills necessary to support computer science education (Alaijlan et al., 2023).

Pedagogy is a vital component of effective teaching and needs to be understood to promote student learning (Kong et al., 2020; Shulman, 1986). Shulman (1986) reminds us that pedagogy alone is not sufficient in regard to teaching and that it must be coupled with content knowledge, wherein he developed the term “pedagogical content knowledge” (p. 6). Pedagogical content knowledge (PCK) includes the comprehensive understanding of both the subject matter and pedagogy, as well as the ways in which they interact (Bjursten et al., 2022). A greater comprehension of the content is necessary to teach some computer science concepts, such as coding. For example, there are implementation strategies, instructional methods, and instructional techniques that are all part of teaching coding (Kandemir et al., 2020). Curriculum integration for computer science will fall within the purview of certified teachers (Northrup et al., 2022). Ensuring all teachers have the training and skills necessary to teach computer science is a key focus for K-12 education (Mertala, 2020). Credentialed teachers must possess the pedagogical content knowledge required to teach computer science effectively; good professional development programs are essential because they will act as the liaison between the standards and implementation (Kong et al., 2020).

Change Initiatives in K-12 Education

Educational systems are often changing to meet new requirements and reforms put in place by policymakers to hold teachers and schools accountable for student learning (Darling-Hammond, 2005). Changes are often initiated in a top-down manner from state legislatures or district leadership ultimately to teachers who will carry out the new initiative (Beycioglu & Kondacki, 2021). There is an assumption by those who mandate these changes that they will also alter teaching practices and lead to positive changes in student learning (Elmore, 1995; Hargreaves et al., 2007). However, change initiatives are not this simplistic, and involve understanding the implementation process, cultivating teacher buy-in, and professional development for effective implementation.

Implementing Change

Darling-Hammond (2005) states, “Change is a constant for educational systems” (p. 362). Education has undergone numerous changes over the past fifty years, making it a commonplace aspect of teachers’ jobs (Hargreaves et al., 2007). Sometimes changes are small and do not require a great deal of effort to implement, while others are more involved, requiring a greater commitment of time and resources. For instance, though they are very simple to adopt and do not require retraining teachers in their practices, structure modifications - like altering the schedule - are seen to have great symbolic value because they communicate a serious message about change (Elmore, 1995). Other changes are more complex using change agents to assist teachers in enhancing their practice by helping them improve instruction (Cohen & Ball, 1999). Implementing change of practice takes time, therefore anyone concerned about effectiveness of policy initiatives must have a thorough understanding of how change occurs in education (Hall, 1992).

In order for teachers to be successful with new teaching methods, policymakers must recognize that the process of change necessitates time and opportunity for teachers to understand the content in a way that enables them to structure and rebuild their practice through in-depth research and experimentation (Darling-Hammond, 2005). Effective implementation is not spontaneous, it demands careful planning and a conceptually robust vision that is organizationally feasible and sensitive to local needs (Fullan & Mundial, 1989). Research indicates that it takes more than two years for an educational innovation to result in the practical or instructional changes that a program is designed to address (Boyd et al., 2009; Brouwer & Kourthagen, 2005; Glazerman et al., 2010; Lee & Min, 2017). Rather than considering the role of teachers as clear-cut and uncomplicated, policymakers need to acknowledge the complexity of classroom conditions that need to be accounted for during the implementation process (Hall, 1992). Therefore, careful planning that accounts for a considerable amount of time for implementation, continuous support for local actors, and addressing ground-level obstacles is necessary for the successful execution of a change initiative.

Change is usually intended to improve instruction and learning (Elmore, 1995). However, change is frequently perceived as a top-down mandate, with legislators forcing changes on schools that seem more like directives from above while ignoring the autonomy required to carry them out (Beycioglu & Kondakci, 2021). Some assume that change itself automatically produces improvements in teaching and learning (Elmore, 1995). However, implementation is not an event, but a process that occurs gradually over time and across systems (Coburn, 2005). Although the top-down approach may have value in kickstarting change, change should evolve during the implementation process (Fullan & Mundial, 1989). Making a paradigm shift helps policymakers focus on building capacity that empowers schools and teachers rather than

establishing controls to direct the system (Darling-Hammond, 2005). Policy decisions and top-down approaches may initiate a needed change but for it to lead to instructional improvement, there needs to be ongoing support for implementation (Fullan & Mundial, 1989; Hall, 1992).

Since they will be leading the charge in implementing new curriculum and teaching methods, teachers must be motivated and thoroughly understand the change initiative in order for it to succeed (Turnbull, 2002; Lee & Min, 2017). Because different teachers will have varying degrees of experience, it is counterproductive to the implementation process to presume that they will all perceive the change in the same manner, which makes support even more crucial (Hall, 1992). By being aware of this, policymakers can concentrate on capacity-building initiatives aimed at increasing the ability of administrators, teachers, and schools to perform duties they have never previously been asked to perform (Darling-Hammond, 2005). Policymakers do not normally have the necessary background for conducting these initiatives themselves, therefore they will need to rely on independent professional development providers, universities, and other experts, to translate policy messages for school districts, administrators, and in-service teachers (Coburn, 2005). Being on the frontline of implementation makes empowering teachers and teacher buy-in critical to the success of the change process (Ertmer, 1999; Darling-Hammond, 2005).

Teacher Buy-in

Students learn more when they are taught by educators who are dedicated to every student learning through clear curricular intentions (DuFour, 2007). In order to do this successfully teachers charged with the responsibility of implementing curriculum need a sense of ownership, or buy-in. Buy-in is “the extent to which individuals believe in and actively support interventions” (Grebing et al., 2023, p. 1). When buy-in is created, there is a greater sense of

ownership because it helps educators create meaning in their work (Bas et al., 2019). For example, Schmoker (2018) identifies the success of Brockton High School's metamorphosis of being the lowest achieving high school in Massachusetts to one of the top 10 percent in the state, to its faculty. The faculty was embarrassed by students' MCAS scores and came together to support their social studies department chair, Susan Szachowicz, who had a plan to raise achievement levels (Schmoker, 2018). While Szachowicz later went on to become the principal of Brockton High School, she was a teacher leader first who was able to garner support and collective buy-in from colleagues giving teachers a renewed ownership of the curriculum leading to increased student achievement. Collective buy-in from educators is an even stronger predictor of successful implementation of curriculum (Grebing et al., 2023).

There are several characteristics that factor into teacher buy-in. Those characteristics include individual beliefs, motivation, self-efficacy, and commitment to action (Haney et al., 2002; Lee & Min, 2017; Macy & Wheeler & Wheeler, 2021; Grebing et al., 2023). For a program to be implemented successfully, teachers must be convinced of its worth and the significance of the effort being carried out (Macy & Wheeler, 2021; Grebing et. al., 2023). One way to do this is to give teachers a seat at the decision-making table. When Seaford, NY wanted to change their reading program, instead of mandating the change, they offered teachers lessons with books from the Bookworms program, the teachers liked the program and a year later it was adopted by the district (Chenoweth, 2020). Clayback (2022) found that the degree to which teachers execute a change initiative is greatly influenced by their beliefs and prior experience. Teachers will feel less control over the change process when they are not involved in it, which is especially true when it comes to top-down change management, as unfavorable teacher attitudes lead to resistance for the change initiative (Beycioglu & Kondakci, 2021). Their practices are

influenced by their beliefs since they have the power to either reinforce or undermine how students react to changes brought about by policymakers (Fives & Buehl, 2016). If teachers are carrying out a program, there must be a collective belief that the change is worth making for them to adopt the change and bring it into their classrooms (Turnbull, 2002; Lee & Min, 2017). Beliefs are also directly connected to motivation because they fuel the desire for change. If one believes a change is possible and has value, consequently, they will find the motivation to make the change (Grebing et al., 2023). As a result, motivation plays a consequential role in implementation (Daritotis et al., 2008).

Self-efficacy is another key component to teacher buy-in; the stronger their sense of self-efficacy, the more motivated they become to implement the changes (Lee & Min, 2017). Self-efficacy is the conviction that one can bring about positive change and achieve the desired results in the classroom (Grebing et al., 2023). Teacher self-efficacy is a crucial motivating factor in determining how much effort is put into professional practice (Ryan & Hendry, 2023). The term “self-efficacy” originated from the research of Albert Bandura (1977) and is derived from his social cognitive theory. Bandura used the term to describe a person’s confidence in their capacity to function successfully and effectively in a particular setting (Ryan & Hendry, 2023; Zhou et al., 2022). Bandura argued that individual self-efficacy was derived from four sources: mastery experiences, vicarious experiences, social persuasion, and emotional arousal (Rich et al., 2017; Ryan & Hendry, 2023; Zhou et al., 2022).

Mastery experiences allow teachers to achieve success through performance activities. For example, when teachers are successful in the implementation of a new instructional approach, their self-confidence grows (Lewis et al., 2021). When teachers see student success it reinforces their ability to deliver on the expected outcomes. With each successive mastery

experience, which is the most powerful of Bandura's sources, teachers build further confidence (Rich et al., 2017; Zhou et al., 2022).

Vicarious experiences like modeling also help teachers gain self-efficacy (Lewis et al., 2021). Modeling provides a sense of security for teachers and motivates them to experiment with new techniques (Ryan & Hendry, 2023). Training teachers to learn by modeling is the foundation of the field experiences required by most teacher education programs. Social persuasion happens when people receive feedback on how they perform on a particular activity (Zhou et al., 2022). Social persuasion increases efficacy when a teacher receives positive feedback, usually from colleagues or administrators, or when working with a coach (Lewis et al., 2022). Teachers are further socially persuaded during professional development by comments they receive on their performance (Zhou et al., 2022). Teachers can increase their self-efficacy by interpreting their level of arousal in difficult situations in a positive way (Ryan & Hendry, 2023). By identifying their level of enjoyment, enthusiasm, and contentment while engaging in an activity, emotional arousal has taken place which also increases their self-efficacy (Zhou et al., 2022). Developing opportunities for teachers to experience all four of Bandura's sources will serve to increase their sense of self-efficacy, which should provide them with the necessary confidence to integrate new subject matter into their curriculum.

Commitment is also important for making long-term and lasting change. A teacher may believe in a program, be motivated to carry it out, and even believe in their ability to do it, but being committed to the success of the program is the final necessary piece. The Success For All (SFA) program provides a full curriculum from PreK through eighth-grade in addition to training and support, with one caveat, teacher commitment – 80% of teachers have to vote, by secret ballot, to adopt the program (Chenoweth, 2020). There is evidence that when a program is first

adopted it may not immediately produce the desired results and that a program needs to mature in order for it to do so (Lee & Min, 2017). This was also the case with SFA in Steubenville, Ohio. Many teachers wanted to abandon the program during its second year, but with a lot of support from administrators, they persevered (Chenoweth, 2020). The more comfortable the teacher becomes with the changes being made and the more knowledge they have about successful implementation leads to success in the long-run. Lee & Min (2017) attribute this to the learning curve of the teacher themselves. A teacher needs to be committed to work through the bumps early in the process of making these changes (Grebing et al., 2023). Steubenville highlights their commitment to the program and after twenty years, still champions its success (Chenoweth, 2020). The four characteristics of buy-in, individual beliefs, motivation, self-efficacy, and commitment to action, work in tandem allowing teachers to realize the curricular goals.

Teachers need to experience some level of ownership of the curriculum that they are teaching. Any modifications to the curriculum must be adopted and implemented by teachers, and if they are to do that successfully, they must be able to comprehend the changes completely (Lee & Min, 2017). Teacher buy-in is an important to any change initiative and its implementation process. It is essential to explore the practices that improve teacher confidence for integrating new subject matter into the curriculum. For computer science to be successfully integrated into the elementary curriculum, teachers' lack of confidence in their understanding of the subject must be addressed (Vegas et al., 2021).

Professional Development

Teacher education and professional development are widely acknowledged as essential components to help teachers adopt and implement new curriculum (El-Hamamsy et al., 2020).

The primary element needed for instruction is a well-trained and knowledgeable teacher and in the absence of that, districts will turn to professional development programs to train in-service teachers to fill the void (Vegas et al., 2022). A one-time, sit-and-get workshop is the most commonly used professional development delivery method, yet this approach to professional development is ineffective regarding influence for changing pedagogical techniques because they lack depth of knowledge and iterative practice (Bigsby, 2017; Whatley & Smith, 2023). Teachers may leave these trainings with some new techniques and abilities, but they might never put them into practice, or only do so in a weak, less integrated way because they do not have the depth of knowledge to feel comfortable with the subject (Lewis et al., 2021; Whatley & Smith, 2023). It is not enough to offer a professional development opportunity, districts must examine what teachers need to know and how to deliver training so that it is adopted into the curriculum in effective ways.

Who will teach computer science has presented a significant challenge to the integration of computer science, especially for elementary education, since there are few computer science graduates in general, and they are not pursuing a career in the teaching profession (Webb et al., 2017). There is a worldwide shortage of teachers qualified to teach computer science (Mason & Rich, 2021; Vegas et al., 2021). As a result, integration of computer science will fall on in-service teachers who may be intimidated by the prospect of delivering content for which they believe they lack the skills to integrate (El-Hamamsy et al., 2020; Oldham, 2021). Teachers will not be able to teach a subject effectively if they do not fully comprehend it (Oldham, 2021). While there is a significant amount of research regarding the integration of technology into teaching practices, there is not an abundance of information available for preparing elementary

teachers to integrate computer science into their pedagogical practices (Ertmer, 1999; Liu et al., 2017; Hutchison et al., 2021; Mason & Rich, 2021; Consoli et al., 2023).

The right training is needed to prepare teachers to teach computer science in elementary schools (Funke et al., 2016). Opportunities for professional development should increase teaching ability in ways that align with the pedagogy (Santo et al., 2019). Teachers' background knowledge, experience, and the amount of time required for them to learn and apply new skills should all be considered when choosing the sort of professional development to be provided (Hestness et al., 2018; Whatley & Smith, 2023). Teachers need time to learn, understand, implement, and then reconvene for further discussion (El-Hamamsy et al., 2020). Teachers find collaborative work helpful to increasing their interest in a subject and training initiatives should provide opportunities for collaborative work (Dorotea et al., 2021). To provide teachers with the best chance for successful integration of computer science into elementary education, professional development should be learner-centered, content-oriented, of sufficient duration, and collaborative to help build the confidence of educators for integrating new knowledge into the curriculum (Dorotea et al., 2021; El-Hamamsy et al., 2020; Mason & Rich, 2019; Spratt, 2019).

Sensemaking

Humans are naturally curious learners, however change is often overwhelming (Willingham, 2021). Many teachers, when faced with change, recoil due to the cognitive load of making the leap from where they are to where they must be, particularly if it is mandated and they were not included in the process (Fink & Stoll, 2005; Hargreaves, 2005). Guidance and professional development may be offered to assist teachers in gaining the necessary skills needed to implement a change, but to impact teaching methods, they must be ongoing (Roehrig et al.,

2006; Darling-Hammond & Richardson, 2009; Archibald et al., 2011). Haney et al. (2002) argued for the inclusion of teacher belief systems in every professional development experience related to change initiatives because those beliefs are the core of the change to be made.

Overcoming resistance, building pedagogical skills, and assessing the development of belief systems that come from a variety of internal and external influences are elements that feed into a teacher's sensemaking for change. These elements ultimately become part of a teacher's sensemaking, an iterative process that is used to understand and implement change (Russell et al., 1993; Pirolli, 2011).

Definition of Sensemaking

Karl Weick (1995), known as “the father of sensemaking”, states that sensemaking is “about authoring as well as about reading” (p. 6; Ancona, 2012; Introna, 2019). This description refers to the dual nature of sensemaking involving both interpretation and construction processes. Making sense establishes a link between the tangible and the abstract (Weick et al., 2005). The term “sensemaking” essentially means figuring out how to make sense of some stimuli, and suggests that understanding requires information processing (Pirolli, 2011). It is how we structure what is unknown or speculated with that which is known or observed in order to guide actions for carrying out tasks, enhancing routines, and making meaning (Klein et al., 2007; Ancona, 2012).

Humans are regularly attempting to make sense of their experience and do so with influence and input from others (Mills et al., 2010). This is particularly true for environments that are rapidly changing (Ancona, 2012). As such, sensemaking is a deliberate and dynamic process of framing and reframing, an ongoing and continuous cycle that leads to deeper understanding (Klein et al., 2007). Sensemaking can also be described in terms of a loop, an

iterative process, that repeats as new information and knowledge is received that reframes previous understandings (Russell et al., 1993; Pirolli, 2011). It is needed most when there is an incomplete understanding of a stimuli and ambiguity exists that requires individuals to determine next steps in a situation (Mills et al., 2010; Ancona, 2012). As individuals struggle with an incomplete reality and strive to create order, they will seek out others for assistance in creating a new framework from which to work (Dervin et al., 2008; Pirolli, 2011; Ancona 2012). In this way, sensemaking is a social activity incorporating data from a variety of resources to create the necessary connections for implementing change (Ancona, 2012).

Sensemaking requires communication (Weick et al., 2005). Part of the definition of sensemaking is the social construction of meaning, which is the process by which individuals get meaning from shared experiences and social interactions (Mills et al., 2010). In situations that require a quick response time and integrating multiple sources of information, or when the complexity of the situation necessitates the use of multiple perspectives and skill sets to understand and make sense of available information, collaborative sensemaking is common (Pirolli, 2011). During sensemaking, people draw on both formal and informal resources, such as chance encounters with coworkers and unplanned talks in public areas like break rooms and corridors (Allen, 2023). In his definition of sensemaking, Pirolli (2011) highlights the social exchanges of knowledge that contribute to the practice of making sense of a new phenomenon. Sensemaking is inherently a collective process, requiring individuals to gather diverse information, listen to various perspectives, and involve others in understanding a situation (Ancona, 2012).

These definitions align with the nature of education, particularly the sense of urgency that is so intrinsic to K-12 education. Legislators pass reforms that require changes, but they do not

identify how those reforms should be implemented, it is up to educators to make those decisions. As a result, reforms may be cloaked in ambiguity requiring sensemaking to bring a sense of order and understanding for those tasked with its implementation. There is an interplay between knowledge, beliefs, and attitudes of these implementers, their circumstances, and the policy signals that makes sensemaking critical in determining meaning for the reform (Spillane, 2002).

Sensemaking in Curricular Adoption

Organizational scientists have proposed sensemaking theory as a way to describe how people learn in work environments, especially in regard to change (Allen, 2023). This may require the acquisition of new subject knowledge, resolving unstructured issues, and developing situational awareness (Pirolli, 2011). Sensemaking allows people to arrive at a thoughtful and useful interpretation of events (Klein et al., 2017). Ultimately, sensemaking transforms people into information designers because they must search for meaning in ambiguity and accept what is possible (Weick, 2005; Dervin et al., 2008). Adopting comprehensive curriculum reform requires fresh collective learning from all reform participants (Soini et al., 2017).

Porter et al. (2015) found that, “The sustainability of any policy rests in whether or not it is able to be routinized or institutionalized within the system” (p. 135). Leaders use their own understanding of a policy, based upon personal experiences and knowledge, in order to decide the best way to implement it (Gawlik, 2015). Those responsible for making sure that implementation takes place throughout a district must engage in intense collaborative sensemaking in order to develop an understanding of extensive curricular change and strategies for managing reform efforts in school districts (Pyhalto et al., 2018). Creating connections between old and new understandings and across educational levels is a key component of shared sensemaking related to extensive curriculum reform (Pietarinen et al., 2019; Lowell et al., 2024).

Additionally, the various lenses that those involved in curricular adoption within their districts use to understand the reform and formulate suitable responses to it are influenced by organizational values, traditions, conventions, and professional practices (Pyhalto et al., 2018). Those who help unpack understanding of the policy also have an impact on district stakeholders' sensemaking (Hill, 2003). Districts consult with what Coburn (2005) calls "nonsystem actors" (p. 24) who may be appointed to state task forces and commissions that provide the necessary resources and are contracted to conduct training related to the state's instructional policy, which is a key component of policy implementation.

School district stakeholders are critical to curricular adoption because they interpret, integrate, and transform the overarching goals the change by encouraging communication and collaboration to improve learning and provide teachers with clear expectations, direction, and support (Spillane et al., 2002; Porter et al., 2015). These district-level stakeholders ensure that the curriculum's goals are effectively carried out by acting as a liaison between the high-level curriculum frameworks, which are often created at the state or federal level, and the practical realities of teaching and learning in local schools (Soini et al., 2017). Collaboration across the district, creates a shared understanding of the reform which benefits the adoption process (Lowell et al., 2024). Shared sensemaking techniques can be used to improve understanding, collaboration, and alignment during the implementation of reform, particularly in a complex system like education, by functioning as tools to help people engage with and actively navigate the reform (Pyhalto et al., 2018). Cristofaro (2022) identified one of the key features of organizational sensemaking being the influence of relationships within an organization, particularly between supervisors and employees. The effectiveness of sensemaking depends on

involving participants in shared sensemaking, which is supported by school leaders who engage with teachers to help them understand signs from their environment (Gawlik, 2015).

At times teachers may encounter competing messages from leaders resulting in ambiguity and uncertainty around implementation (Allen & Penuel, 2015). Thus, sensemaking is a means of addressing or resolving these uncertainties and ambiguities (Stollman et al., 2020). The process of sensemaking is inherently social, drawing on resources from coworkers and supervisors to create a collective understanding of the significance of the reform (Coburn, 2001; Weick et al., 2005; Mills et al., 2010; Ancona, 2012; Luttenberg et al., 2013; Siciliano et al., 2017; Lowell et al., 2024). People use dialogue and social contact to create common meanings and interpret information from their surroundings, which Gawlik (2015) found to be an important aspect of teacher sensemaking. The messages that leaders give teachers about how to take part in the reform and the messages that teachers receive from other institutional actors gives the reform its meaning (Allen, 2023; Lowell, 2024). Allen (2023) found that teachers rely significantly on others' knowledge of local expectations as part of their sensemaking process. Teachers' interactions are largely shaped by their social networks, leading to a convergence of their beliefs with those of their colleagues (Siciliano et al., 2017). By using accessible cues from their environment, sensemaking helps teachers eliminate ambiguity and uncertainty and arrive at understanding of the reform (Allen, 2023). Whether educators accept, reject, or adopt curricular reform is rooted in sensemaking (Pyhalto et al., 2018). Engaging people in sensemaking is a critical component to the success of curricular adoption (Gawlik, 2017).

The way in which reform concepts and practices are integrated into instruction is greatly influenced by the sensemaking processes that teachers employ (Allen, 2023). Most reforms are designed to improve student achievement, and although the mandating body expects teachers to

carry them out as intended, once implemented, they do not always work out as planned (Stollman et al., 2020). Because sensemaking is retrospective, curricular adoption is also impacted by teachers past adoption experiences, both successes and failures (Seligman, 2006). Fullan (2016) admits that, while doable, long-term sustained change is hard to accomplish, particularly at scale, when involving new skills or knowledge. Requiring teachers to implement policies that they do not understand undermines meaningful engagement and only serves to promote compliance (Elmore, 2016). Teachers often disregard policies when they do not grasp certain aspects of them (Coburn, 2001). Even when they are confident in their own abilities to carry out the reform, they nevertheless recognize that time and resource constraints may necessitate deviating from the recommended course of action (Siciliano et al., 2017). When it comes to significant improvements involving numerous stakeholders in an educational system, shared sensemaking can be successful (Soini et al., 2017). Since teachers are at the center of educational reform, their participation in sensemaking is essential to attaining the desired results (Schmidt & Datnow, 2005).

Sensemaking in Education

Sensemaking is described as an ongoing process that fills important gaps in organizational theory occurring most often when something is different from the way that it is expected to be (Weick et al., 2005). This occurs in education when a change takes place requiring educators to implement an innovation and change a practice (Stollman et al., 2020). For teachers, this is the intersection between their perceptions of what the situation calls for and their personal frames of reference (Luttenberg et al., 2013). Ultimately, a teacher's sensemaking arises from the demands of their situation combined with their perception of those demands; these elements are crucial in the sensemaking process for educators (Stollman et al., 2020).

The role of individual cognition is significant in the sensemaking process for teachers. The ability of teachers to interpret new information about teaching and manage changes to their work practices is influenced by their personal, social, and material resources (Allen, 2023). Prior knowledge, experiences, and individual belief systems are involved in the construction of understanding for a stimulus or change initiative (Spillane et al., 2002). Sensemaking is done to provide a more comprehensive image and broaden teacher understanding of the change as they work toward implementation (Klein et al., 2007).

In education, sensemaking is triggered by a change initiative, and is most pronounced where there is ambiguity regarding the change initiative. Often these change initiatives are introduced through a new state policy. Teachers initially interpret these new policies in a variety of ways, through messages from state and local governing bodies, curriculum and standards, and outside influencers, such as those who provide professional development to educators (Coburn, 2002, Coburn, 2005, Allen & Penuel, 2015). Sometimes these sources of understanding provide too many interpretations leading to confusion by the educators tasked with implementing them (Allen, 2023). To stabilize their sensemaking, educators will therefore turn to classifying and organizing their experiences (Weick et al., 2005).

Sensemaking in schools is an iterative process that engages teachers in meaning and develops their capacity for implementation. Teachers rely upon prior experience with ambiguity as a way to move forward with understanding (Allen, 2023). Additionally, new understanding draws upon prior knowledge to create new meaning (Spillane, 2000). Teachers will then interpret policies and put them into practice based on that prior knowledge, all of which contributes to the development of a frame of reference in regard to the reform (Spillane et al., 2002, Allen & Penuel, 2015). The frame is a reflection of the total experiences one has, thereby contributing to

making sense of a situation or phenomenon (Klein et al., 2007). Communication, therefore, plays an important role in the organization of sensemaking because much of how we make sense of the world is based upon how we interact or communicate within it (Weick et al., 2005). There are a multitude of actors that overlap one another with whom teachers communicate and who influence the process of sensemaking in a variety of ways. Professional development is one such way in which teachers begin to make sense of a reform (Luttenberg et al., 2013). Policy ideas are often translated to teachers through professional development providers both within and outside of the school system, playing a crucial role in the implementation of policy and adding a new layer of knowledge to a teacher's existing understanding (Coburn, 2005). Professional development about policy should include information about standards, which also play a critical role in the sensemaking of teachers because the standards are typically at the heart of a reform and how the policy reform is expected to manifest itself in student learning (Heredia, 2020, Allen, 2023). At the district level instructional guidance is provided to teachers through district trainers or coaches (Coburn, 2005). Finally, there is a formal and informal social aspect to teacher sensemaking. Teachers will engage in casual conversations with peers that discuss and explore others' ways of sensemaking around reforms. They will also learn from administrators, through a social context, about expectations and understanding around implementation of the reform (Ancona, 2012, Allen & Penuel, 2015). These conversations allow them to find information and deepen their learning about reforms (Weick, 2005, Pirolli, 2011). Sensemaking in education is "about creating an emergent picture that becomes more comprehensive through data collection, action, experience, and conversation" (Ancona, 2012, p. 6). Ultimately, sensemaking around reforms allows educators to develop understanding that is adaptive and

refined over time as they contemplate a change initiative through their own frame of reference and the external factors that influence them.

Summary

There is not a single, widely recognized definition of computer science, but it is crucial for teachers to understand that it is its own distinct subject (Tucker, 2003). Just as important is for teachers to understand that computer science is not fundamental computing skills or technology integration (Fluck et al., 2016; Wohl et al., 2017; Oldham, 2021). Beginning in the 2024-2025 school year, the state of Tennessee has implemented a change initiative that expects elementary general education teachers to incorporate computer science into the curriculum.

Change initiatives are typically driven by the desire to improve student learning (Elmore, 1995). However, they do not guarantee better teaching or learning outcomes, which will need ongoing support for effective implementation (Fullan & Mundial, 1989; Hall, 1992). There are several factors that affect implementation, among which include capacity building activities, teacher buy-in, and ongoing professional development (Darling-Hammond, 2005; Bas et al., 2019; El-Hamamsy et al., 2020). To properly implement the change initiative, teachers must have a thorough understanding of it (Turnbull, 2002; Lee & Min, 2017).

Sensemaking involves organizing knowledge by connecting known and unknown elements together (Weick, 1995). It involves both interpretation of policy, internal and external influences to construct new meaning (Spillane et al., 2002). Sensemaking is an iterative process that repeats with the introduction of new information leading to deeper understanding (Russell et al., 1993; Klein et al., 2007; Pirolli, 2011). Sensemaking is a social activity that involved integrating information from an interacting with a range of sources to establish the connections required to carry out change (Weick, et al., 2005; Ancona, 2012).

Therefore, the purpose of this qualitative study is to explore the perceptions and experiences of elementary general education teachers regarding their capacity to integrate computer science into their curriculum. The study aims to investigate how teachers make sense of a new curricular initiative – in this case, computer science integration – what factors influence that sensemaking, and how sensemaking impacts their assessment of their own preparedness to carry out a change initiative. The findings will offer insightful information to support primary education-related professional development and teacher preparation initiatives.

The literature review supports this study's aim to understand how teachers interpret the introduction of computer science into the elementary general education curriculum. It creates a framework for understanding computer science and its adoption into the curriculum. It explores the implementation of change initiatives and the elements that impact and challenge change. It investigates the external influences of teachers as they make sense of their capacity to incorporate computer science into the elementary general education curriculum. This qualitative study will include data from interviews to gain insight into their sensemaking. The following chapter will explore the method of the study, justify its purpose, and explore the framework in correlation to the research presented in Chapter 2.

CHAPTER III: METHODOLOGY

METHODOLOGY

Research Purpose and Questions

The purpose of this qualitative study was to explore the perceptions and experiences of elementary general education teachers regarding their capacity to integrate computer science into their curriculum. The specific goal of this study was to investigate how teachers interpreted the integration of computer science into their curriculum, what factors influenced their perception of their readiness to implement a change initiative. This research analyzed teachers' attitudes and experiences in order to identify factors that contributed to their sensemaking to teach computer science concepts, providing valuable insights that can inform professional development and teacher training programs pertaining to elementary education.

This study will address the following research questions:

1. How do elementary teachers understand and experience computer science integration into their general education curriculum?
2. What factors influence teachers' perceptions of their preparedness for integrating computer science into the general elementary education curriculum?
3. How do teachers perceive their preparedness for integrating computer science into the general elementary education curriculum?

Theoretical/Conceptual Framework

This study leaned on sensemaking as its theoretical framework. Instead of focusing on how judgment affects decision-making, sensemaking explores the interaction between action and interpretation (Weick, et al., 2005). Sensemaking theory has been applied by education researchers to analyze how teachers have responded to new policies and programs that have been

implemented in their schools (Allen & Penuel, 2015). Spillane et al. (2002) used sensemaking when examining how educators interpret and implement new policy initiatives. Allen & Penuel (2015) investigated the ways in which sensemaking influenced their pedagogical choices and determined that, “Successful implementation of new standards will require focused attention to teachers’ sensemaking and the development of supports that help teachers make sense of ambiguous situations and manage uncertainty” (p. 147). I used sensemaking to examine how teachers made sense of a change initiative through the interplay between their knowledge of computer science, external influences and their perceived preparedness related to teacher capacity for integrating it into their curriculum. Figure 1 demonstrates my methodology for how teachers use sensemaking to understand and implement change. The change initiative (depicted in green) was introduced to teachers within a thought bubble to represent the new policy requiring the integration of computer science into the general education curriculum. The thought bubble also signifies that the initiative continues to evolve as new information is synthesized. The change initiative interacts with teacher’s prior knowledge of computer science and their professional learning experiences, marking the initial stage of interpreting the change. External influences (shown in blue) include the policy, standards, messages, and guidance introduced throughout the iterative process, supporting and guiding the teacher’s sensemaking as they assess their capacity to implement the change. Ultimately, this process will result in the integration of computer science into the general education curriculum.



Figure 1: Theoretical Framework

Subjectivity/Positionality Statement

Being a child of the '70's I distinctly recall the guest speaker coming to my elementary school to teach us how to use the Tandy TRS-80 and giving us the *Whiz Kids* comic book to take home. Our exposure was limited to one day, however. In my later elementary years, I was exposed to what we call pair programming today, this is where students collaborate with a partner to use programs. In pair programming, one student operates as the driver by being hands-on with the computer while the other is the navigator, helping the driver make decisions about programming and identifying mistakes in the code (Code.org, 2014). When I was in school, we used pair programming for such programs as Lemonade Stand and Oregon Trail on the Apple II. My exposure to computer science was through computer integration which was typical of most of my generation.

I began using computers daily right after college as a graduate assistant. Not only did I have to learn how to use a DECmate desktop computer for daily responsibilities, but I also had to teach myself how to use the Apple computer in the office to create a desktop blotter sized calendar using PageMaker that advertised all of the events sponsored through the office of student activities for each month of the academic year. The assistant director of student activities was very familiar with computer technology and software and she showed me some basics, gave me the software manuals, and I took it from there. That became the common standard of practice for me throughout my career. Whenever I needed to use a computer for something new, I sought out someone knowledgeable for guidance and relied on user manuals for the rest.

The world wide web gained prominence as I began my professional career. At that time, the college I worked for offered a course teaching employees to build department websites using HTML code. I participated in the training for our department but left the position before creating

the department webpage. I did not use those skills until I began working for a higher education non-profit consortium several years later. The power of the internet had become mainstream and computer science concepts had begun to seep into jobs that had never needed them before. Web management became part of my job description and while the software was considered a WYSIWYG (what you see is what you get) platform, there were some parts of the website that did require understanding programming and writing code. Once again, I found myself seeking out my own resources and consulting with professional web developers and turning to Google to help me write HTML code for the website.

During those years, I was also a busy mom who volunteered regularly at my children's school, initially teaching typing skills to fourth grade. Typing instruction soon morphed into teaching weekly computer technology lessons to grades 3 – 6 at a small parochial school. Initially, I focused on technology integration but over the course of the eight years that I taught there, I developed a curriculum that encompassed both technology integration and computer science principles. This came about because of my own professional experiences which began to rely more on problem-solving skills and understanding how computer programs worked in order complete tasks such as creating event registration forms and conference and event apps for smartphones. It was clear to me that more people would require computer science skills, and I did not want my students to be at a disadvantage. I researched, attended Code.org workshops, and established relationships with computer science professors at a nearby college in an effort to expose my students to the subject. I also completed an M. Ed. conversion program to earn my teaching certification which led me to accept a full-time position as a K-5 computer science teacher in an urban school district. I soon realized that I had limited exposure teaching K-2, so I spearheaded my own professional development once again and grew my personal library with

books and resources to support computer science education at the elementary level. I even embarked on a one-year certificate program through Tufts University to study early childhood technology.

My experience learning to be a computer science teacher is not unique, in fact, elementary school computer science teachers often come from different disciplines and report being self-taught. Many teachers in elementary schools are from generations that did not have in-depth use of computers or access to computer science courses throughout their K-12 experience. Some may have been exposed to computers, but with no pathway to an endorsement, these teachers, like myself, had to seek out their own training and professional development to be prepared to introduce computer science concepts to their young students.

The new Tennessee law requiring computer science be integrated into the general elementary education curriculum will require teachers to unpack a new set of standards, ones that they have had limited or no exposure to previously, and develop understanding of a new subject matter and how it dovetails with what they are teaching in other content areas. As stated, elementary school computer science teachers had to seek out opportunities to grow their knowledge to become content area experts, but these were teachers who were intrinsically motivated to do so. Developing knowledge for a new content area, particularly computer science, requires a great deal of time to read, attend workshops, and explore computer science concepts. For in-service teachers tasked with the responsibility for integrating computer science, time is something that they do not always have for such exploration and may not be something they are willing to commit to doing if they are not personally interested in the subject matter.

Viewpoint of the Study

Bhattacharya (2017) stated, “Constructivist studies examine how participants form meanings and actions, and get as close to the experience as possible.” (p. 105) with the goal to reveal what is valued, believed and assumed by the researcher and participants. The epistemology of this qualitative study was constructivism because it sought to construct meaning from the perceptions of general education elementary school teachers’ regarding the integration of computer science into the curriculum (Crotty, 1998). I explored the “experiences of individuals” through an open-ended interview format (Creswell, 2009, p. 113). This facilitated an interpretive dialogue, a distinguishing feature of a constructivist study, between the interviewee (teacher) and myself as the researcher to understand teachers’ perceptions (Bhattacharya, 2017).

Interpretivism served as the paradigm for this study because it looks to make sense of the world in which we live and work (Creswell, 2009). Interpretivism is centered around in-depth understanding of individuals and their interactions with the world (Bhattacharya, 2017). The meaning individuals make of their world is based upon the perspective of the individual being studied (Crotty, 1998). The interpretivism paradigm of this study was chosen because its goal was to comprehend how teachers’ perspectives are formed with reference to their preparedness to incorporate computer science into the elementary general education curriculum. It was an interpretive inquiry about what each teacher heard, saw, and understood (Creswell, 2009).

Research Design/Methodological Approach

A fundamental qualitative study, grounded in constructivism and often termed interpretive, is the most popular type of qualitative research in fields like education, and in the absence of a specific study type (phenomenological, grounded theory, ethnographic, etc.), it can be labeled as basic (Merriam & Tisdell 2016). This study was a basic qualitative study that

utilized in-depth interviewing to construct meaning (Marshall et al., 2021; Merriam & Tisdell, 2016). The qualitative interview is a crucial tool for investigating how participants perceive and understand their surroundings by offering a window into the subjects' world as they articulate their thoughts, feelings and experiences in their own words (Brinkmann & Kvale, 2018). Through interviews, qualitative researchers can identify the meaning frameworks that people employ to structure their experiences and make sense of the world (Hatch, 2002).

This study used a formal, in-depth, semi-structured interview protocol. Planned interviews are formal and enable the researcher to delve further into the teacher's perceptions, leading to a greater depth of understanding (Hatch, 2002). Formal interviews such as these follow a pre-arranged set of questions that serve as a guide but are semi-structured because they also allow for additional questions that may come up during the interview process (Marshall et al., 2021). This type of interview creates knowledge through the interviewee-interviewer exchange and subsequent analysis of qualitative data (Brinkmann & Kvale, 2018).

I used an interviewer-interviewee exchange to collect data regarding sensemaking around this change initiative. This study sought to investigate how elementary general education teachers made sense of their capacity to integrate computer science into the elementary general education curriculum. A teacher's perception of their ability to carry out a change initiative depends on a variety of elements once it is presented. Teacher perceptions of their capacity is not a phenomenon that can be easily observed or studied over long periods of time. Additionally, it cannot be easily quantified by collecting hard data. According to Brinkmann & Kvale (2018), "The qualitative research interview is a construction site for knowledge" (p. 8). Conducting a qualitative interview is like working on a construction site. Every question is a tool, selected with care to mold the framework of understanding. The interviewee's responses serve as the

foundational elements required to assemble a story. The researcher creates and modifies the framework, assembling the components into a logical whole. While buildings are constructed from plans on a construction site, knowledge and understanding of human experiences are gained through qualitative interviews. In order to accomplish this effectively, a researcher needs to speak with the interviewee directly to understand their perspective and the elements that influenced how they made sense of their capacity to implement change. It was essential to this study that I communicate with interviewees, and one-on-one interviews conducted in-person, by telephone, or virtually were the most effective ways to accomplish this.

Research Site

The research was carried out in a large school system situated in a rapidly growing area of Tennessee (National Center for Education Statistics, 2022). With a diverse student body, this urban school system consisted of 160 elementary, middle, and high schools. About 60% of students came from low-income homes and two-thirds were English language learners. The district emphasized the importance of professional development for its educators by offering ongoing professional development opportunities, year-round, and allotting each educator ten professional development days of their choosing, each year.

The district participated in a state sponsored train-the-trainer initiative in February 2023, aimed at equipping educators and districts to integrate computer science into the curriculum. The district computer science leaders later brought together teams from each elementary school, in the summer of 2023, to provide their own train-the-trainer program, the *Policy to Practice* (P2P) training. Elementary school principals were asked to assemble teams of four that included one K-2 grade classroom teacher, one 3-5 grade classroom teacher, one instructional coach, and one representative from the school's administration team. Team members were labeled *Integration*

Engineers and attended the P2P training to learn more about computer science and how to integrate it into the general education curriculum. Their role was to serve as school-based instructional leaders, supporting teachers in the implementation of computer science and collaborating with the district's computer science team to deliver ongoing professional development within their school. The P2P training introduced the state policy (Appendix C), the district's implementation roadmap (Appendix B), new computer science state standards, offered hands-on activities that included computational thinking principles, and explored resources and lesson plans for computer science. Teams worked together to create an implementation plan for integrating computer science into the general education curriculum at each elementary school, which served as a deliverable to the district. As part of the implementation plan, *Integration Engineers* offered training throughout the 2023-2024 school year to prepare teachers for full computer science integration in Fall 2024. Each team also received a box of resources for their school that included books, posters, and robots.

Each of the 70 elementary schools in the district were required to send a team of four people to one of the five P2P training sessions offered. All schools that offered computer science as part of their general education curriculum also designated their computer science teachers as members of the *Integration Engineers* team. They also attended the P2P training sessions and were actively engaged in the implementation of computer science instruction at their respective schools. Including both groups of teachers made it possible to compare the sensemaking processes across different teaching contexts. The district was selected for this study because it trained 280 *Integration Engineers* as part of the computer science initiative, providing a deep pool of potential participants for this district-wide study.

Since qualitative research is not designed for mass participant interviews, a process for participant selection must be used to find suitable candidates (Glesne, 2011). Therefore, a purposive sampling was determined to be the most effective approach for this study. This kind of sample is used because “they are information rich and illuminative, they offer useful manifestations of the phenomenon of interest” (Patton, 2002, p. 40). Purposive sampling is used when it is necessary for participants have direct experience with the topic under investigation (Creswell, 2009). Participants in this study were elementary school teachers who were identified by attendance at the P2P training in the summer of 2023. The district required that this researcher collaborate with the computer science department to identify and nominate potential participants for the study. As a result, this researcher requested P2P training attendance records and contact information from the district’s computer science department for all elementary school educators who attended the district’s P2P training.

Following the identification of potential participants, this researcher sought principal approval prior to initiating contact with teachers, in adherence to district requirements. Principals were provided a list of teachers identified from the P2P training records and were asked to grant permission for those teachers to be contacted. Only teachers whose principals approved participation received invitations to take part in the study. Teachers who expressed interest in participating in the study responded to the email invitation, and interviews were scheduled accordingly. Five non-computer science teachers and five computer science teachers were interviewed.

This researcher had prior professional relationships with some of the participants through professional development activities. Recognizing that these relationships could potentially influence interpretation, deliberate steps were taken to minimize bias. Prior to conducting

interviews, I reflected on these relationships and considered how they might influence the research process, including advantages and disadvantages of including the participants. To minimize interviewer bias, I adhered closely to the interview protocol, used open-ended questions, limited unrelated conversation, and maintained a neutral demeanor throughout. Member checking was also employed to ensure the participants' perspectives were accurately represented.

Data Collection

When we cannot observe how individuals interpret the world around them, conducting interviews becomes necessary to understand their thoughts (Merriam & Tisdell, 2016). Ten interviews were conducted either in person, by telephone, or via a web conferencing platform, in accordance with the interviewee's preference. Each interview was conducted in August 2025 and ranged from 30 to 90 minutes in length. Field notes captured researcher observations and reflections during data collection. All interviews were recorded for transcription purposes. These recordings will only be available to the researcher and will be destroyed five years after the completion of the study.

To ensure a precise constructivist interpretive study, open-ended interview questions were designed to explore the "experiences of individuals" (Creswell, 2009, p. 113). A semi-structured interview allowed the researcher to guide the conversation toward the specific information being studied and provided the flexibility to modify the questions asked and probe other topics that arose from the interview process (Merriam & Tisdell, 2016; Bhattacharya, 2017). Semi-structured interviews were appropriate for this study because it investigated how teachers made sense of a change initiative and the external influences that influenced their sensemaking. Semi-structured interviews reveal meaningful patterns that are difficult to spot and

usually concealed from direct observation (Hatch, 2002). Since each teacher interviewed had a unique set of experiences, their sensemaking was developed in ways that had to be discussed in order to be fully understood, which is why semi-structured interviews were a useful method. The interview was guided by a protocol to ensure continuity between interviews (Creswell, 2009; Merriam & Tisdell, 2016).

Weick's (1995) three aspects of sensemaking – identifying two elements, such as past experiences and current challenges, and comprehending the relationship between these elements to construct meaning – were used to formulate ten questions created for use in the semi-structured interviews in order to learn about the sensemaking process teachers employed when integrating computer science into their curricula.

This study sought to answer three questions regarding the integration of computer science elementary general education teachers:

1. How do elementary teachers understand and experience computer science integration into their general education curriculum?
2. What factors influence teachers' perceptions of their preparedness for integrating computer science into the general elementary education curriculum?
3. How do teachers perceive their preparedness for integrating computer science into the general elementary education curriculum?

The questions used in the interview were designed to address several topics: understanding policy, defining computer science, exposure to computer science, integration of computer science, factors that influence teachers' perceptions, and preparedness for integration. I

was be able to evaluate the teacher's understanding of computer science and the implementation requirements by asking questions regarding policy and terminology. The ability of teachers to incorporate computer science was assessed through several key questions that provided context about their specific situations. Contextualization provided me with the opportunity to frame questions designed to better understand the participant's unique circumstances and experiences that influenced their behavior (Hatch, 2002). By examining their previous experiences with integrating computer science, any challenges they encountered, and their self-evaluation of their readiness for the task I was able to gauge how prepared they felt to integrate it into the curriculum (Weick, 1995). Understanding a teacher's prior knowledge, the messages they received about implementation, and any additional computer science training they had provided insights into the factors that shaped their sense of readiness to teach computer science. Questions addressing these aspects are crucial for analyzing the elements what influenced a teacher's perception of their preparedness to implement computer science.

Data Analysis

The process of data analysis creates meaning of the open-ended data collected by a researcher (Creswell, 2009). Inductive analysis was used in this study because it does not rely upon a predetermined premise that must be verified or refuted through data collection (Bhattacharya, 2017). Inductive analysis is a research method that focuses on open questions rather than testing deductive hypothesis; categories and relationships are derived from evidence, leading to the development of theories (Tuckman & Harper, 2012). Interviewees' statements must be compiled, condensed, and interpreted in order to make sense of the data (Merriam & Tisdell, 2016). The process of analysis is iterative, necessitating many transitions between different phases and procedures for the researcher (Bhattacharya, 2017). The process that I used

to analyze data included reviewing interview transcripts several times, decomposing the data in multiple rounds of coding, identifying categories and themes from the data, and synthesizing the data. Creswell (2009) referred to this process as basic qualitative analysis.

I began by reviewing interview transcripts several times to increase readability and develop a deeper understanding of the material. Superfluous words like “uh” and “um” were removed to produce clean transcripts. Each coding cycle employed the use of software intended to easily organize transcription excerpts. Clean transcripts were uploaded to Delve software where initial codes were assigned. Following this stage, the clean transcripts were uploaded to Dedoose software where the coding process was repeated. The process yielded over 800 coded excerpts. Immersing myself in the data enabled me to recognize the main points made by the interviewees and provided me with an understanding of the overall richness and complexity of the dataset (Creswell, 2009). An open coding approach was used to deconstruct the data into manageable segments, with phrases, sentences, and paragraphs highlighted content relevant to the study’s research questions (Merriam & Tisdell, 2016; Bhattacharya, 2017).

The first round consisted of descriptive coding. Descriptive coding summarizes the primary topic from a portion of qualitative data (Saldaña, 2013). The passage was summarized using single words or succinct statements that were relevant to its topic. The coding process was repeated several times based on each interviewee’s transcript. Once all the transcripts were analyzed, the next cycle included axial coding where categories begin to emerge (Saldaña, 2013). Merriam & Tisdell (2016) define axial coding as, “the process of relating categories and properties to each other, refining the category scheme” (p. 229). The single words and phrases from the previous phase were grouped together during the axial coding process based on commonalities that were categorized.

An additional procedure and software were employed to support theme development. The coded data was imported into Lucidspark which allowed for the clustering and visualization of related codes. The data were grouped based on patterns that emerged as relevant to the research topics. Pattern coding was then used to identify major themes from the categories that surfaced (Saldaña, 2013).

Two types of questions were posed by this study. Because the first research question focused on understanding computer science integration, which is best represented through thematic analysis, categorical pattern coding was an appropriate analytical approach. The additional questions focused on teachers' perceptions, which were best illustrated through explanatory pattern coding. Finally, the data was synthesized to create meaning around teacher capacity for integrating computer science into the curriculum.

Trustworthiness and Rigor

Given that the main goal of this qualitative study was to achieve understanding, it was essential to apply criteria to ensure its trustworthiness (Merriam & Tisdell, 2016). A study must follow specific guidelines to validate its integrity and give readers confidence in its findings (Creswell, 2009). Guba (1981) recommends four criteria for judging trustworthiness: credibility, transferability, dependability, and confirmability. The criteria include multiple measures that the researcher uses to confirm the trustworthiness and rigor of the research and its conclusions (Shenton, 2004).

The first criterion to consider is if the findings are credible, does the study measure what it is meant to measure? There are several strategies that can be used to measure the credibility of the study (Guba, 1981). Shenton (2004) identifies several of these strategies that were utilized in

this study including: using established research methods, familiarity with the organization's culture, participant integrity, iterative questioning, and member checks.

Established research methods were identified and implemented with fidelity. These methods, which directed every aspect of the study, included specific questions, a theoretical framework, and data analysis procedures. This study outlined an interview protocol that included questions intentionally selected to address the research questions that guided the study. For this particular study, a theoretical framework was created to clarify and direct the research. In order to assess a teacher's ability to integrate computer science into the curriculum, this framework considered the introduction of a change initiative (computer science integration), identified prior knowledge, explored attitudes and beliefs toward technology and change initiatives, and recognized the external influences that shaped teachers' sensemaking. Data analysis, like other studies, played a significant role in this study and is evidenced by well-defined coding techniques that involve many iterations of inductive analysis using descriptive, axial, and pattern coding.

The research site and its culture should be familiar to the researcher (Shenton, 2004). This study's research site was identified as a large school system in the United States and located in a rapidly growing section of Tennessee. The researcher had a prior relationship with this site and was familiar with its operations and culture. In addition, the researcher contacted the computer science coordinator for this district to discuss the nature of the study and potential benefits of its findings.

This study outlined a deliberate process for participant selection. Including a broad range of participants enhanced the trustworthiness of the study by reducing bias and ensuring findings were not skewed toward the experiences of a single group. A purposive sampling method was

employed to select participants who had all completed the same P2P training. The sample included both veteran and novice computer science teachers, teachers with computer science endorsements, and non-computer science teachers.

Shenton (2004) stated that strategies should be employed to address the integrity of the participants. To that end, this study included an interview protocol that was designed to assist in developing rapport and educate participants that they could opt out of the study at any time. This study's protocol included specific wording that explained the purpose of the study, the level of anonymity that could be expected by a participant, their rights as a participant to withdraw from the study, and option to review notes and themes developed from their individual interview data.

Iterative questioning is another strategy that can be included to contribute to participant integrity (Shenton, 2004). Another way this study ensured participant integrity was through open-ended questions. These questions were created to guide the interview, including probes that helped to create an iterative process. The interview protocol included questions that overlapped, for example, one question asked about messages participants may have received regarding integrating computer science into their curriculum versus their experience in doing so. The overlapping of these questions allowed the researcher flexibility to follow up on any contradictory statements made by a participant.

Guba (1981) highlighted the significance of member checks to verify the accuracy of assertions made by participants. A member check is the way in which the accuracy of qualitative data is determined based upon returning research findings to participants to verify them (Guba, 1981; Shenton, 2004). To verify patterns and themes that emerged from the analysis, I shared a summary of each participant's interview data for their review. Member checking is another credibility check because it validates the findings of the researcher (Saldaña, 2013).

The second criterion, transferability, refers to the generalizability of the results from one study to the next (Merriam & Tisdell, 2016). Qualitative studies are often limited in their ability to be extrapolated more generally because they are typically small and specific, but transferability is not impossible as it may be generalizable to another organization with a similar situation (Shenton, 2004). Employing the use of thick descriptions will allow other organizations and individuals to identify similarities and determine transferability to their own situation (Merriam & Tisdell, 2016). In the case of this study, teachers' perceptions of their ability to implement a change initiative, specifically integrating computer science into the general elementary curriculum, offered significant transferability because it connects with various research topics and their contexts.

Dependability, the third criterion, is the degree to which study findings can be replicated (Creswell, 2009; Merriam & Tisdell, 2016). Shenton (2004) indicated that replication relies upon a detailed methodology, design of the study and its execution. Dependability is demonstrated in this study through documentation of participant selection, data collection procedures, interview protocols, and data analysis. Guba (1981) recommends overlapping methods as another tactic to promote dependability. The overlapping methods in this study included participant interviews, probing questions, participant reviews of interview summaries, and iterative coding conducted across multiple software applications.

The final criterion outlined by Guba (1981) is confirmability. Confirmability in qualitative research is the impartiality of and freedom from researcher bias (Miles et al., 2013). Measures must be implemented to ensure that findings reflect the views of the participants rather than researcher preferences or biases (Shenton, 2004). The researcher included a subjectivity statement in this methodology section to bring into the open potential researcher biases.

Confirmability can also be measured by using triangulation, which involves gathering information from multiple sources, as previously mentioned regarding the credibility criterion (Guba, 1981). Multiple sources were used to strengthen the credibility of the findings through triangulation. District training materials were also incorporated to corroborate the interview data. The district training materials included PowerPoints used during the P2P training that helped clarify the agenda and training activities.

Conclusion

This basic qualitative study aimed to explore teacher sensemaking around a change initiative, in elementary general education classrooms. It explored how teachers understood and experienced computer science integration into the curriculum, the factors that influenced their perceptions of preparedness, and how they perceived their preparedness for implementation. The next chapter will highlight the findings of the study, looking at themes that emerged from the data.

CHAPTER IV: FINDINGS

Introduction

The purpose of this qualitative study was to explore the perceptions and experiences of elementary general education teachers regarding their capacity to integrate computer science into their curriculum. General education teacher was defined as all teachers, including special content area teachers (e.g. music, art, etc.), who are part of the formal academic program and instruction of all elementary students. The overall goal was to gather data on teachers' sensemaking of a new curricular initiative, computer science integration. I sought to explore how they understood and experienced computer science within their curriculum, the factors that influenced and shaped their sensemaking, and how sensemaking impacted their assessment of their preparedness to implement a change initiative.

Data Collection

For this basic qualitative study, I collected data through semi-structured interviews, as this method provided the flexibility needed to capture participants' experiences and perspectives while those experiences were still unfolding. I conducted the interviews in-person, by telephone, virtually, in August 2025. I utilized the transcription services embedded within the video conferencing software and edited them in their downloadable format, Microsoft Word. I coded the interviews using two different qualitative data analysis software programs, Delve and Dedoose. I also participated in memo writing and kept a reflexivity journal. The research questions that guided my study were as follows:

RQ1: How do elementary teachers understand and experience computer science integration into their general education curriculum?

RQ2: What factors influence teachers’ perceptions of their preparedness for integrating computer science into the general elementary education curriculum?

RQ3: How do teachers perceive their preparedness for integrating computer science into the general elementary education curriculum?

Participant Profiles

Ten teachers participated in this study, selected through purposive sampling. Participants were identified based on their attendance at the *Policy to Practice* (P2P) training held in June 2023, and their contact information was obtained from official attendance records and the district email directory. Of the 120 people contacted via email, sixteen expressed interest in participating, with ten agreeing to be interviewed. All participants were asked the same or similar questions from the Institutional Review Board (IRB)-approved interview protocol. I adjusted the interview protocol for participants who had not implemented any computer science-related lessons. In addition, follow up and clarifying questions were posed to each participant as needed. Participants represented ten different elementary schools across the district. To ensure confidentiality, I assigned each participant a pseudonym. Table 1 provides information about each participant, their teaching role, professional computer science instruction experience, and computer science endorsement status.

Table 1

Overview of Participant Roles, Computer Science Experience, and Endorsement Status

Name	Teaching Role	CS Instruction Experience	CS Endorsement Status
Ayla Larkson	5 th Grade Teacher	Novice	Yes
Tate Stone	4 th Grade Teacher	NA	No
Cam Chase	5 th Grade Teacher	NA	Yes
Lindsey Kerr	Computer Science Teacher	Veteran	Yes

Table 1 continued

Name	Teaching Role	Computer Science Instruction Level	Computer Science Endorsement
Morgan Lovette	4 th Grade EL Co-Teacher	NA	No
Adrian Locke	Computer Science Teacher	Novice	Yes
Blair Wilder	Computer Science Teacher	Novice	Yes
Dylan Clarke	Kindergarten Teacher	NA	No
Sloan Wells	Computer Science Teacher	Novice	No
Logan Hughes	Computer Science Teacher	Veteran	Yes

Ayla Larkson had more than seventeen years of teaching experience in both elementary and middle school. Larkson described herself as a self-directed learner and believed it to be her responsibility to understand and adopt new initiatives, rather than wait for administrative direction. This perspective was driven by her commitment to ensuring that students have equitable access to 21st century skills to remain relevant. Larkson did not have any formal education in computer science; however, she completed the state's computer science pathway to earn the endorsement. Larkson's understanding of computer science is shaped by prior experiences integrating technology into the curriculum, which included teaching keyboarding. She also served as a STEAM/STEM teacher where computer science was a component of the instruction. As a self-directed learner, Larkson actively sought ways to integrate computer technology into teaching by researching software and programs that enhanced academic lessons and expanded students' skills.

Tate Stone was a mid-career teacher with twelve years of experience teaching first through fourth grade in two urban districts. In addition to a bachelor's degree in education, Stone possessed two master's degrees, one in urban education and the second in curriculum and instruction. Although Stone has no formal education in computer science, she believed her training as a general education teacher has prepared her to connect new ideas to prior knowledge

allowing her to continually build and expand her expertise. Stone viewed this background as an advantage when implementing new initiatives and that this would also be the case with computer science implementation.

Cam Chase is a 21-year veteran teacher with experience in both middle and elementary school, though she preferred working with elementary students and was teaching fifth grade. Chase described herself as a “digital immigrant,” recalling the days when loading a single website could take so long that one could leave and return before it finished. Her first computer was a TRS-80, a reminder of how much technology has evolved over time. Chase holds a master’s degree in curriculum and instruction with an emphasis on technology. She was committed to exposing students to computer science, emphasizing that students are growing up in a digital world and must have equitable access to technology to stay relevant. Her understanding of computer science closely aligned with technology integration, though Chase notes that computer science extended beyond simply using computers. Chase regularly incorporated technology into her teaching and had recently earned the K-12 computer science endorsement.

Lindsey Kerr was a veteran teacher with thirty years of teaching experience. Kerr began teaching in private schools before moving to public education sixteen years ago. Over the years, Kerr has taught a wide range of grade levels, from Pre-K through 8th grade math, and had been teaching computer science for the past nine years. While Kerr did not have an extensive background in computer science, she took computer-related courses during both her undergraduate and graduate studies. Kerr self-described as an educator who is always evolving – constantly seeking new ways to approach instruction and researching strategies to enhance learning. Kerr embraced change in education and resisted the monotony of teaching the same

content in the same way year after year. When she began teaching computer science, she viewed her role as helping students learn how to use computers and be successful with basic digital skills. However, Kerr's perspective shifted in 2023, when new state computer science standards were introduced. Afterward, Kerr emphasized deeper thinking and problem-solving, encouraged students to move beyond the surface-level skills, and engaged with computer science as a tool for critical thinking.

Morgan Lovette is an experienced educator with eight years of teaching across multiple elementary grade levels. In the 2025-2026 school year, Lovette was an English Learner (EL) co-teacher in fourth grade. Using computers came naturally to Lovette, who also held a master's degree in instructional design and technology. Lovette recalled feeling successful in both teaching and student learning during the COVID-19 pandemic, when instruction was conducted online, viewed technology as an asset in her practice. While she recognized that computer integration and computer science are not the same, Lovette saw the two as closely related and overlapping. Lovette's personal definition of computer science aligned closely with the state of Tennessee's definition.

Adrian Locke was an 18-year veteran teacher and served K-8 students with special academic and behavioral needs. After several years as a classroom teacher, Locke became an instructional specialist and coach but returned to the classroom in the 2025-2026 academic year as a new computer science teacher. Teaching computer science brought together two passions: teaching and technology. Locke believed computer science instruction should focus on using technology as a meaningful tool rather than a flashy distraction. Taking a blended approach, Locke integrated computer science with STEAM education to create engaging lessons while intentionally limiting students' screen time. Locke had little exposure to computer science prior

to completing the state's online endorsement course, which he did successfully. Locke had also participated in technology-focused professional development and took a portion of a Python coding course. Locke believed it was essential for students to be equipped to use technology as a problem-solving tool. His understanding of computer science, shaped by the endorsement course, emphasized it as a way of thinking, using computational thinking to address challenges. Locke saw computer science as a pervasive field, influencing everything from engineering to mechanics, and believed students should be prepared to navigate a world where it is embedded in nearly every profession. Most importantly, Locke believed that the learner should always be the main focus, not the computer.

Blair Wilder was a veteran teacher with eight years of experience. Wilder studied early childhood education, earning certification to teach pre-kindergarten through fourth grade. In college, Wilder considered majoring in computer science but ultimately chose teaching. When the opportunity arose to teach computer science, Wilder eagerly embraced it, seeing it as an opportunity to have the best of both worlds, combining a passion for technology and teaching. In the 2025-2026 school year, Wilder was in his third year of teaching computer science. Wilder credited his father, a mathematician with a master's degree in computer science for sparking his love of computers. Growing up surrounded by technology gave Wilder a deeper understanding of computer hardware and software than many of his peers, fostering an early sense of confidence with computers. His first real exposure to computer science concepts, however, came through the state's endorsement course, which Wilder successfully completed.

Dylan Clarke was a veteran educator with twenty years of experience teaching and leading intervention groups for students in grades K-6, with most of those years dedicated to teaching kindergarten. Clarke's bachelor's degree was in K-6 education and she also had an

English Learner (EL) certification. She described herself as a professional that embraced growth, took risks to strengthen her practice, and actively sought new initiatives to enhance student learning. Clarke identified as a teacher who consistently integrated technology into her teaching practice but recognized that computer science extends beyond simple technology use.

Sloan Wells is a second career teacher with eleven years of experience, teaching third through fifth grades. While Wells has taught all core subjects, more recently her focus was on 5th grade science and math. This year Wells has begun an exciting new chapter as a dedicated computer science teacher for students in kindergarten through fifth grade. Wells was enrolled in the state's computer science endorsement pathway, at the time of her interview, and was completing a master's degree in mathematics, which also included eighteen hours in learning technology with curriculum design. Her goal was to help students see the computer as a powerful tool to enhance learning. Wells encouraged students to become creators of content rather than merely consumers.

Logan Hughes was a second-career teacher with over eighteen years of experience whose background as a music therapist shaped her student-centered approach to teaching. Beginning as a music teacher, Hughes developed a passion for integrating technology which led to a transition into adaptive technology supporting students with significant disabilities. Her role later evolved into educational technology and, most recently, computer science, reflecting a sustained commitment to innovation and accessibility. Hughes' knowledge of computer science was initially grounded in familiarity with the standards, and much of her expertise has been developed through trial and error. She identified her greatest strength as modifying curriculum and selecting tools that make learning accessible for all students. Having completed the computer science endorsement course, Hughes noted that her earlier exposure to the field was limited.

Reflexive Journal Notes

During the interviews and coding process, I maintained a reflexive journal to record my thoughts, observations, and analytic reflections. This journal served as a space to document insights about participants' experiences, as well as to critically examine my own analytical decisions throughout the coding process (Saldaña, 2013). Reflexive journaling helped me capture emergent ideas, identify early thematic patterns, and enhance the overall credibility and trustworthiness of the study. It also complemented the interview data by deepening my understanding of teachers' experiences with computer science integration and prompting me to consider ways implementation efforts might better support teachers with limited computer science experience.

The reflexive journal directly informed several methodological decisions, including choices related to coding categories, theme refinement, and my dual role as a researcher and a computer science teacher. For example, the category, *Sense of Competence*, emerged through repeated journal reflections. Originally coded as an individual concept, separate from others, further reflection revealed that it was closely tied to confidence, self-awareness and teachers' perceived preparedness. This led me to elevate it to a broader category that aligned more clearly with the research questions.

Similarly, the theme *Instructional Guidance* was refined through the journaling process. I initially referred to it as *Ensuring Instructional Alignment*, but reflection revealed that "ensuring" implied a guarantee that policy, standards, and P2P training alone could not provide. Drawing on Cohen (1995), I reconsidered the language to more accurately reflect that they support but do not ensure instructional alignment. This reframing strengthened the conceptual clarity of the theme.

Ultimately, the reflexive journal allowed me to examine my positionality with greater intentionality. As described in Chapter 3, I entered this study with a prior relationship with this district, and I wanted to remain aware of how that insider perspective could influence my analysis and interpretations. Writing in the journal allowed me to question myself and my own actions. The journal allowed me to question my assumptions, monitor moments of potential bias, and ask whether I was representing participants' perspectives accurately rather than through my own understanding of district practices. This ongoing reflexive questioning guided decisions such as relying on verbatim quotes and incorporating artifacts from the P2P training in evidence. These practices supported neutrality and helped ensure that participant voices, not my prior knowledge, shaped the study's findings.

Interview Data Analysis

The data I collected was coded in cycles from which categories were formed and themes emerged (Table 2). Because it does not rely upon a predetermined premise, inductive analysis was used to identify the relationships and categories derived from the data which led to the development of theories (Tuckman & Harper, 2012; Bhattacharya, 2017). Creswell (2009) calls his process basic qualitative analysis. Organized by research question this section reveals the findings from the interviews and coding processes.

The interview data revealed five themes demonstrating teachers' sensemaking in regard to the computer science integration change initiative. The first theme, *Understanding Begins with Core Concepts*, emerged from the basic understanding and definitions each participant had of computer science. *Understanding Begins with Core Concepts*, demonstrates a starting point for teachers regarding their understanding of computer science. Conceptual structures are ways in which information is organized by individuals seeking to understand the world around them

(Blandford et al., 2014). As such, definitions can be considered an early entry into conceptual knowledge around a topic (Ross, 2001). The definition of computer science reveals a teacher's understanding of the content area as they engage with or embark upon a new initiative. The second theme, *Instructional Guidance*, related to understanding both the legislative intent behind policy and the standards outlined for guidance in implementation often through district training. This theme emerged from the participants' concerns about understanding the state expectations regarding policy and how to incorporate the standards into practice in their classrooms. Those responsible for implementation must first interpret the meaning and intent of the policy in order to adapt the curriculum accordingly (Spillane & Callahan, 2000). Furthermore, standards play a critical role in curriculum development by ensuring instructional coherence and alignment, underscoring the importance of teachers' understanding of the standards they are expected to apply in instruction (Allen & Penuel, 2015). The third theme, *Practical Realities of Computer Science Implementation*, emerged from the varied approaches used to integrate computer science, despite their shared experiences through the P2P training and *Integration Engineers* program or completion of the state's endorsement pathway. Spillane & Callahan (2000) found that reform ideas that are familiar to adopters tend to generate greater engagement than those perceived as ambiguous or unfamiliar, which helps explain the differences in how participants approached implementation. *Both External & Internal Factors Shape Teachers Perceptions of Preparedness* emerged as the fourth theme. This theme became evident based upon the participants' explanations of how they came to know and understand the computer science initiative through their own knowledge and experiences versus external messages from others and trainings. When engaging in sensemaking, our mental models of understanding improve when we involve others in our sensemaking (Ancona, 2012). Finally, through identifying a sense

of competence, our participants exposed the fifth and final theme, *Building Self-Efficacy*. This theme is a self-assessment of participants' personal confidence regarding their sense of preparedness for integrating the new initiative. Participants identified a variety of factors with which they engaged and indicated that their confidence grew as a result of these engagements. Educators that engaged with ongoing external factors begin to change their internal views and continue to improve their sense of competence for implementation of a change initiative (Rich et al., 2017; Ketelhut et al., 2020).

Table 2

Principal Coding Practice (Saldaña, 2013)

<i>Initial Codes</i>	<i>Categories</i>	<i>Themes</i>	<i>Theory</i>	<i>Research Question</i>
Understanding of CS; CS Definition	Knowledge & Definitions of CS	Understanding Begins with Core Concepts	When teachers demonstrate understanding of the core concepts of computer science, they are able to distinguish computer science from other technology-related concepts. This allows them to implement computer science more effectively and with greater fidelity, rather than equating it with technology integration or digital literacy.	RQ1
Understanding Policy; Understanding Standards	Guidance	Instructional Guidance	When teachers develop an understanding of policy and academic standards they are better prepared to implement instructional changes.	RQ1
Current Implementation; Challenges & Barriers to Implementation; CS Integration	Implementation & Integration	Practical Realities of Implementation	Teachers interpret and respond to computer science implementation as an ongoing process. They implement in different ways based on their beliefs and the challenges they navigate; implementation can lead to varied outcomes.	RQ1

Table 2 continued

Professional Development; Messages Received; Tech	External Factors; Internal Factors	External & Internal Factors Shape Teachers' Preparedness	Teacher preparedness is shaped by the complex interaction of both external and internal factors.	RQ2
---	------------------------------------	--	--	-----

Attitudes; Change				
Attitudes; Self-Directed Learning				
Self-awareness, Confidence, Disposition, Training needs	Sense of Competence	Building Self-Efficacy	Building self-efficacy requires teachers to continue engaging with external and internal factors to grow their own sense of competence for their ability to integrate CS into their curriculum.	RQ3

Research Question 1

How do elementary teachers understand and experience computer science integration into their general education curriculum?

This question explored the background and knowledge teachers possessed regarding computer science. It elicited information about the participants' professional preparation, their definitions of computer science, the guidance they received related to policy and standards, and their experiences with implementation and integration. From this question three themes emerged that illustrate how teachers understood and experienced computer science integration. These themes are: *Understanding Begins with Core Concepts*, *Instructional Guidance*, and *Practical Realities of Computer Science Implementation*.

Understanding Begins with Core Concepts

Understanding Begins with Core Concepts demonstrates a starting point for teachers in relation to their understanding of a new initiative like computer science. These core concepts include defining computer science and understanding what it encompasses. When asked to define computer science, most participants offered definitions that reflected their existing knowledge and understanding of the subject. Most of the definitions were similar, aligning with all or part of the definition outlined in the *Computer Science Integration Guidebook (TN DOE, 2023)* – problem-solving, computational thinking, and foundational concepts. Others, however,

directly correlated with the use of computers through digital literacy and technology integration. This theme reinforced the notion that establishing a shared definition of computer science is challenging (Oldham, 2021). However, the variation among definitions may reveal commonalities upon which a shared understanding can be built.

Table 3

Non-Computer Science Teachers' Computer Science Definitions (endorsed and non-endorsed)

<i>Computer Science Definitions</i>	<i>Quote</i>
<i>Thinking/Problem Solving</i>	"When I think of computer science, I think of problem solving. I think of things like variables and code and some things that are way up here [holding hand above her head]. Really using technology to problem solve."
<i>Computer Use</i>	"The study of computers, technology, how it works, how it integrates into our world, in the different facets of life, and in the classroom."
<i>Components</i>	"Being really knowledgeable about the hardware and the software and what you're getting ready to implement."

Table 3 presents the definitions offered by non-computer science teachers, with or without a computer science endorsement. While the definitions differ, common throughlines emerged, such as the emphasis on a working knowledge of technology, understanding its role in problem solving, and recognizing its practical applications.

Table 4

Computer Science Teachers' Computer Science Definitions (endorsed and non-endorsed)

<i>Computer Science Definitions</i>	<i>Quote</i>
<i>Thinking/Problem Solving</i>	"I really enjoy the definition where it brings in the whole thinking part, problem solving and with the power of computers, because that together just encompasses everything to me."
<i>Computer Use</i>	"Teaching kids to be content creators and using that to enhance their education, instead of content users."

Definitions from computer science teachers (see Table 4) also share some similarities. Again, while the definitions vary, there are common throughlines such as an emphasis on

thinking, viewing computer science as a tool that empowers learners, and recognizing its role in creation. Both groups of participants shared similar throughlines, recognizing the cognitive component of computer science through its emphasis on thinking and problem-solving, its application as a tool for accomplishing tasks, and its functional role in fostering innovation.

The state of Tennessee cited a computer science definition by the Association of Computing Machinery as “the study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation, and their impact on society” in their Computer Science Integration Guidebook (2023). However, Tennessee expands that definition to emphasize that computer science is more than technology integration or the use of computing devices. Computer science includes how people think and work with computational tools and is a way for students to “express their ideas and apply their creativity to solve a variety of different problems” (TN DOE, 2023, p. 10). The similarities between definitions may be attributed to the P2P training where many were introduced to it for the first time.

Another important core concept that emerged from this study is teachers’ understanding of computer science. Like definitions of computer science, there are multiple ways in which participants demonstrated their initial understanding of computer science and at times it may be easier to understand computer science through what it isn’t (Oldham, 2021).

Multiple participants expressed their understanding of computer science through the use of computers as a tool in education, while also noting that they understood that it is “more than just computers.” Additionally, many teachers identified a starting point for their understanding as the integration of technology, regardless of their endorsement status and the P2P training explanations. In fact, even veteran computer science teachers pointed to the P2P training and

computer science endorsement course as a turning point for them in fully understanding computer science.

Kerr, a veteran computer science teacher, described herself as always evolving. When she first began teaching computer science, almost a decade ago, she viewed her role as helping students learn how to use computers and develop basic digital skills. She described her early understanding of computer science in relation to technology integration, explaining, “When I first started this nine years ago, the whole entire concept of computer science was not on my radar” and “It was more of how to survive on the computer rather than how to think and how to program and things like that.” However, after the state revised the computer science standards her perspective shifted. She now emphasizes deeper thinking and problem-solving, encouraging students to move beyond surface-level skills and engage with computer science as a tool for critical thinking.

Chase, a fifth-grade math teacher, described herself as a “techie” who has always been comfortable with computers. She participated in the district’s P2P training program and has a computer science endorsement. Believing that we live in an increasingly digital world, she wants students to know how to effectively use the technology available to them. By the end of the first week of school, she ensures that, “they know how to share their screen, know what the Windows key is. I have enrolled them in a typing course, and they are learning how to use their e-mail.” Throughout the year students complete additional assignments to strengthen their digital literacy skills. “They’ve done PowerPoint. They had to read a book online to a kindergartner and record themselves using Canva,” she explains. It wasn’t until she began the computer science endorsement course that she was introduced to topics such as coding. Now, she incorporates

coding activities into her teaching, including projects where students create algorithms and program Bee Bots to spell out their names.

Clarke, a kindergarten teacher, shared, “I was involved in using computers and technology since the moment I stepped into the classroom; it definitely has evolved.” She points to COVID as a major influence on the increased use of technology in her teaching, noting that using computers became a primary focus during that time. Although she tries to incorporate aspects of computer science into her curriculum, she admitted that she is not always comfortable with it. “The most experience I have with kindergartners and technology is them actually using it and doing things with when we were logging in and going to iReady and Lexia,” she explained. After attending the district’s P2P training, Clarke began intentionally integrating computer science vocabulary into her classroom routines, as recommended in the training. For example, she describe classroom procedures as algorithms, explaining to her students that it is a “computer word” that means doing things step-by-step.

Wilder’s passion for computers was nurtured early in life by his father, a mathematician with a Master’s degree in computer science. Having spent much of his life around computers, Wilder developed a depth of background knowledge uncommon among his peers, even building two computers from scratch. Although he had long considered pursuing a degree in computer science, he ultimately chose a career in education. When a computer science teaching position opened a few years ago, he saw it as the perfect blend of his two interests and eagerly stepped into the role. Reflecting back to his first year, Wilder described it as “iffy” explaining that while he was familiar with computer science, he quickly realized that his prior knowledge “can only get me so far.” He acknowledged that, at first he “didn’t always think beyond just using computers. I didn’t always think as deeply.” However, his perspective shifted as he became more

involved in computer science through the endorsement program and attending the district's P2P training.

Regardless of endorsement status or engagement with the P2P training, teachers consistently identified basic computer skills as their initial entry point into computer science – even among those serving as dedicated computer science teachers. When teachers demonstrated an understanding of the foundational or core concepts of computer science, they were better able to distinguish these ideas from general technology skills. For many, these core concepts served as a natural starting point for integrating computer science into their existing curriculum.

Instructional Guidance

The phrase “instructional guidance” describes the ways in which the state and local education systems use frameworks, standards, policies, and professional development to try and guide and support teaching and learning (Cohen, 1995). Education policy is inherently a top-down process originating at the state level and passed through districts to administrators and ultimately to teachers (Beycioglu & Kondakci, 2021). The successful implementation of policy depends on the ability to shape both individual and collective actions toward achieving a common goal (Coburn, 2016). Teachers encounter policy in multiple ways - through standards they are expected to teach and through professional learning designed to inform and guide implementation (Coburn, 2005). Effective implementation requires support from various levels of the system, particularly from the school district itself (Hall, 1992). Moreover, teachers develop their understanding of innovative teaching and learning strategies from a wide range of sources - people, organizations, and policies, that are frequently tangentially related (Coburn, 2005).

In the case of computer science integration, the policy was originated at the state level, and the state partnered with a technology-based non-profit organization to provide training for districts. The district in this study subsequently developed its own training, *Policy to Practice* (P2P), for teachers designated as *Integration Engineers*. These teachers were responsible for training colleagues at their schools to implement the policy and standards. As part of the P2P training, teachers received a copy of the state’s computer science policy (see Appendix) and schools were required to create implementation plans outlining how computer science would be integrated into instruction. Participants demonstrated awareness of the policy’s expectations and identified areas where further clarification was needed (Table 5).

Table 5

Knowledge and Understanding of Policy

<i>What is your understanding of the policy regarding integration of computer science into the curriculum and the state standards?</i>	<i>“My current understanding is that students have to have some type of integration with all content in their classes.”</i>
	<i>“I know it is a policy. That is the extent of my knowledge, but I do know that it is a policy and that we are required to teach these things.”</i>
	<i>[laughter – mouths “don’t remember”] “I remember going to a training two years ago and we played with robots. I’m sorry.”</i>
	<i>“My understanding is that it is supposed to be embedded in all curriculum for elementary school, middle school is supposed to have one class, and high school is supposed to have an entire year.”</i>
	<i>“Very little understanding, if at all.”</i>

Participants’ knowledge of policy was strongest among those serving as computer science teachers, regardless of endorsement status. Although the P2P training artifacts reflected that the computer science standards were unpacked during sessions, participants described differing levels of understanding. For example, Lovette noted a need for more information, explaining, “If I had to teach computer science, I don’t know what standards we need to do.” In

contrast, Larkson indicated that she was able to deliver a lesson directly from the standards. Some participants, like Locke, critiqued the standards as being too easy and suggested they should be “a little bit more ambitious.” Chase demonstrated both understanding and uncertainty, stating, “Some of these standards, like ‘used advanced features of digital tools’, I’m on that,” but added, “Some of them I have not figured out.” She further reflected that if she knew the computer science standards as well as the math standards, she could be more intentional in her implementation.

Scope and sequence serves as a fundamental framework for teaching, providing concrete guidance for implementation and ensuring alignment with state standards. It functions as a roadmap for teaching and learning, outlining what is taught and when. Participants expressed differing levels of awareness regarding the computer science scope and sequence, with some emphasizing its value and others noting its absence. Wilder described using the scope and sequence to plan weekly lessons, stating, “It is definitely nice having an idea of where I’m going, what the sequence is.” Conversely, Locke, a new computer science teacher, indicated that having a scope and sequence would be helpful. Chase echoed this sentiment and explained that she is currently developing her own. Kerr, who serves on the Computer Science Steering Committee, discussed helping to create a scope and sequence that identifies essential standards for inclusion, particularly when rotation schedules limit instructional time.

The district’s role in professional development further supports the theme of *Instructional Guidance*. Although the district provides numerous training opportunities, the P2P training was intentionally aligned to the state’s computer science policy. This initiative served as the primary point of entry for teachers as they began implementing the new standards. Nearly all participants

described the P2P training as beneficial in helping them understand and engage with the computer science initiative (Table 6).

Table 6

Relevance of Training

<p><i>How relevant and helpful was the training?</i></p>	<p><i>“I would say it shapes my understanding. It’s kind of the foundational part, that beginning part of it.”</i></p> <p><i>“The initial training formed my understanding. It put it into context and talked about how you could implement it. That’s what I needed as a new computer science teacher.”</i></p> <p><i>“The training was definitely relevant, because you have to start from somewhere, they were helpful as a starting place.”</i></p> <p><i>“It was very interesting, They were very engaging but as far as direct application, I struggled to make the connection.”</i></p> <p><i>“Very helpful, very relevant.”</i></p> <p><i>“There was really great information. I remember we did a cool activity where you had a robot go north-south-east-west, which is really cool.”</i></p> <p><i>“That training was good in that they said, ‘Hey this is coming down the pike, be ready for it, get a little plan together.’”</i></p>
--	--

Practical Realities of Implementation

Implementation is not a single event but a gradual, systemwide process that unfolds over time (Coburn, 2005). Teachers initially understand change in terms of themselves and what they, individually, think it entails (Hall, 1992). As they engage with implementation, teachers discover challenges that shape how the implementation process unfolds, leading to differing levels of implementation across a district (Clayback, 2022). This theme became evident through teachers’ descriptions of the decision-making processes that guided their curricular decisions.

We interpret incoming information through the lens of what we already know (Willingham, 2021). Similarly, the teachers in this study approached implementation through the framework of their prior knowledge and experiences. Wells describes her initial experience

teaching computer science in familiar terms - through vocabulary instruction - explaining, “I’ve always taught vocabulary.” She also viewed her role as one of support, helping homeroom teachers by equipping students with foundational digital skills. For example, she teaches students to use ClassLink and the learning management system because, as she noted, “That takes another job away from the teacher.”

Other teachers, however, did not see clear connections between their curriculum and computer science. Stone indicated that the curriculum does not “lean towards it” and as a result, she doesn’t “lean a lot towards it in class.” While she found the training to be engaging, she noted that it “lacked the connection, the real-world application.” Ultimately, she believes the best opportunity for integration lies in teaching vocabulary. Wilder, on the other hand, described a lesson he created using Minecraft and ciphers – an idea inspired by his own enthusiasm for the game. “I love Minecraft,” he said, “so anything where I can interact with them playing Minecraft is great for me.”

Teachers encounter numerous challenges implementing a change initiative, including issues of buy-in, access to adequate resources, and time. Clayback (2022) found that gaining teacher buy-in is critical at the outset of an initiative, while Kincaid (2007) identified faculty buy-in as a key factor in the implementation of an initiative. Similarly, teacher buy-in surfaced repeatedly in participant interviews, both directly and indirectly. Wells described the difficulty she experienced with one team’s reluctance to embrace her role as a computer science teacher. She recounted, “Ninety-five percent of my teachers are super supportive and they’re so happy for me to come.” However, one team expressed skepticism about her efforts to teach students to log onto their computers, questioning, “They don’t know their letters. How are you going to do

that?” Reflecting on this experience, Wells admitted, “I wish I would have a little bit bigger buy-in from those teachers.”

Stone, Lovette, and Clarke each questioned the feasibility of implementation and teacher buy-in due to restrictions placed on their other curriculum. They especially feel that their English Language Arts (ELA) curriculum is inflexible and that they are not allowed to add or change what they are teaching. Stone clarified that educators must adhere to the curriculum and utilize all available resources, including PowerPoint presentations, without altering them. “How are we supposed to integrate those things if we aren’t allowed to change?” she asked. Similar sentiments were echoed by Lovette, who noted that the curriculum’s pacing was difficult, saying, “It’s very tight. It’s tight [with emphasis]. They won’t be prepared for the benchmark if you stray from it, even for a single day.” Lastly, Clarke expressed uncertainty about how to reconcile integration with the message of faithfully adhering to the curriculum.

“I, for one, and maybe others really feel this intense pressure day-to-day. The district expects fidelity and they come in to observe you with the book open. We are told fidelity means we are not adding to it, but then integration of things like this initiative is something that is expected. How do those two things balance? How do those two things work? I don’t understand how that will work.”

Both Kerr and Wilder were disappointed by colleagues’ lack of enthusiasm when they were tasked with launching the initiative at their school. Kerr explained the effort and her role as its lead during a meeting with the faculty. “I made computer science posters that they could hang in their classrooms,” she said, “Honestly, not very many people hung them up and it all still falls on me to implement.” According to Wilder, the largest obstacle to this endeavor is lack of teacher buy-in. Like Kerr, he talked about fruitless meetings with faculty members to discuss how they could implement computer science in their classrooms and what was being done to

support it. He encountered some educators who were resistant to new ideas and have been teaching in the same manner for a long time. They question, “Why do I need to add computer science?”

According to Turnbull (2002), implementation requires sufficient resources. Several participants addressed the importance of resources for implementation. Lovette recalled testing robots in the P2P training and receiving one or two to use in the classroom. One issue that she explained is that there was only one robot to share among a small group and only thirty minutes of available instructional time. Wells also mentioned that her students lacked resources. She was given only one cart of computers to share among seven kindergarten classrooms, meaning that one laptop was available for every 3.5 students. It is not feasible to share one computer with multiple students because of the way in which students must sign into the computer. Because of the method that students must use to log in, sharing a computer with several students is not practical.

Hughes is concerned with how difficult it is for her special needs students to gain access to resources. In her position, she modifies the curriculum to accommodate her students’ needs. As a result, she frequently has to make manipulatives from scratch to provide students to access computer science. Despite the software investment, she says that the applications often “bury them so far down that no one can ever find them” or that “it is really not accessible without a whole lot of environmental modification because they are not designed for that.”

Finally, Kerr expressed concerns about the inequitable funding between computer science and other related areas. Despite the expenses related to a computer science curriculum, the funding is not available for computer science teachers. “I only have my BEP money to spend,” she said, “because I didn’t get a whole lot of money from my school.” Whereas others get a lot

of money from their department. However, she does express gratitude for the Computer Science Library where teachers can borrow a class set of Bee Bots, Spheros, and other equipment.

Theorists indicate the need for time to be allocated to the implementation of new initiatives (Fullan, 1989; Hall, 1992; Kincaid, 2007; Joram et al., 2020). It takes time for teachers to prepare for implementation of new curriculum, and they should have the necessary time to do so (Shin et al., 2021). While the state’s intention was not to add another thing to teachers’ already full plates, the participants did not see it as intended, a complement to what they were already teaching. They saw it as another initiative requiring instructional time. Many participants in this study referenced the need for a commitment to more time for implementation and instruction (Table 7).

Table 7

Challenges Encountered with Integration

<p><i>What challenges have you encountered trying to integrate computer science into your curriculum and what were the biggest obstacles?</i></p>	<p><i>“I think lack of time would be the biggest obstacle. I think if you gave every teacher robots and training and got them really excited, they’d say there’s no way I can fit this in.”</i></p>
	<p><i>“I know it really eats up time. I wish we had the time to integrate more.”</i></p>
	<p><i>“I think that we already cannot complete successfully 100% of what we’ve already been given to complete. Adding any kind of instruction on top of that, even integrated into it as far as computer science, time is a big factor in that.”</i></p>
	<p><i>“There just needs to be time allocated.”</i></p>
	<p><i>“Even though I have all these great resources, I haven’t had the time to really sink my teeth into any of them.”</i></p>

Kerr discussed the concerns regarding implementation and time with students. Schools that had a computer science teacher added the subject to their related arts rotation, meaning students attended class based upon a rotation schedule. However, that was also difficult because there were so many different schedules among schools. Sometimes classes were seen on a 6 or 7

day rotation while other schools that saw classes on a 10-day rotation. This created an uneven implementation even across schools with computer science in the related arts rotation.

Summary of Research Question 1

This research question explores how teachers understand and experience computer science integration. Participants described their understanding of computer science as an evolution from basic skills to more complex problem-solving, beginning with what was familiar and accessible, typically foundational computer skills. Analysis of participant interviews reveals a developing understanding of core computer science concepts, reflecting a continuum from novice to veteran that illustrates how understanding is constructed over time and mirrors the iterative process of sensemaking.

Participants also emphasized the importance of professional learning opportunities that introduced them to the state teaching standards as well as the district's implementation plan and their role as *Integration Engineers*. Those who demonstrated a clearer understanding of the policies and standards were better equipped to navigate implementation challenges than those that did not. Ultimately, the question highlights implementation as a systemwide process that occurs slowly, over time, as Coburn (2005) observed. The participants' experiences reflected the iterative nature of sensemaking – beginning with what they knew, building upon it, and reframing their understanding each time they encountered new information.

Research Question 2

What factors influence teachers' perceptions of their preparedness for integrating computer science into the general elementary education curriculum?

This question explored the external and internal influences that helped shape a teacher's perception of preparedness to implement the change initiative. It identified external factors such

as professional development, teacher networking, and messages received from outside actors. The question also looked at the attitudes teachers have regarding change and technology initiatives, and their self-directed research. From this question one theme emerged, *External and Internal Factors Shape Teachers' Preparedness*.

External & Internal Factors Shape Teachers' Preparedness

Teacher preparedness represents a symbiotic relationship between external and internal factors. Understanding is shaped by the interaction of internal and external factors and is understood through prior experiences, ideas, and knowledge (Bandura, 1986; Weick, 1995). This theme emerged from the participants interactions with external factors related to implementation and the attitudes they held regarding technology and change initiatives.

External factors identified by participants centered on professional development, teacher networking, and the messages received from the district, administrators, and colleagues. Professional development encompassed several forms, including district-sponsored professional learning, networking opportunities, and district leadership teams. Only the dedicated computer science teachers attended computer science related professional learning programs outside of the P2P training, even though they were open to all teachers in the district. Most expressed overwhelmingly positive views and high levels of satisfaction with the professional learning offered by the district related to computer science.

Wells attended Planning Days trainings offered and available to all teachers, and explained that most of her guidance for teaching computer science came from those workshops. She described them as, "some of the better trainings that I've been to." Kerr also took advantage of these opportunities and participated in every training offered. Her involvement led to an appointment to the Computer Science Steering Committee, where she collaborated with other

computer science teachers to design and facilitate the Planning Days workshops. Wilder also attended several trainings and found Minecraft 101 and 201 particularly helpful for integrating computer science concepts such as algorithms, sequencing, and ciphers. Locke, however, only learned after the start of the 2025-2026 school year that there was an in-service each semester for computer science teachers. Prior to that, he attended STEM trainings but found that they offered limited sessions related to computer science.

Those who were not dedicated computer science teachers reported only attending the P2P training. While many participants expressed appreciation for the experience, several wished that additional sessions had been offered, describing it as “one-and-done” training. Clarke reflected, “I assumed that maybe that group would be called back to come the next year and talk about next steps, what it might look like, but we did not. So that was it, unfortunately.” Similarly, Stone noted that the P2P training was “the only experience I’ve had receiving training with computer science.” Both Larkson and Chase described the P2P training, followed by their computer science endorsement course, as the extent of their professional learning in computer science.

Several computer science teachers emphasized the importance of networking opportunities in their professional learning. Participants engaged with two forms of networking, formal and informal. Informal networking often occurred through one-on-one interactions or informal gatherings of computer science teachers. Wilder shared that when he began teaching computer science three years earlier, he connected with other teachers over lunch during a district in-service and those conversations helped him acclimate to his new role. Wells also reached out to other computer science teachers to exchange ideas and share best practices as she prepared to become a computer science teacher. In contrast, Hughes described experiencing periods of isolation due to the unique nature of her position, expressing, “It would be nice if I

didn't feel like an island all of the time," and adding "the best support I can give is to surround myself with people who are at least willing to come and visit the island."

Three participants - Hughes, Kerr, and Wilder - serve on the Computer Science Steering Committee, which they describe as a form of formal networking since the group meets regularly to plan and develop trainings for other teachers. They credited their involvement on the committee as a positive influence on their preparedness to teach computer science (Table 8).

Table 8

Formal Networking Attitudes

<i>Kerr</i>	<p>"I go to my friends on the Computer Science Steering Committee quite a bit because they make me feel a whole lot better about certain things."</p> <p>"I love the Computer Science Steering Committee because we bounce ideas off of each other, so I feel like that is important."</p> <p>"Also, we can learn from the high school and middle school computer science teachers on the team, and they can learn from us."</p>
<i>Hughes</i>	<p>"I have been part of the Computer Science Steering Committee since its inception, which has given me more knowledge, otherwise I probably would not have a whole lot of that."</p> <p>"And now I am part of the Computer Science Steering Committee. It's been nice being able to see even more in-depth with so many other educators that all feel the same way I do about this learning."</p>
<i>Wilder</i>	<p>"They just all love it [computer science], and it's awesome being around other people that love it."</p> <p>"It's been a confidence booster that we can get there."</p>

In addition, Wells and Locke expressed their satisfaction with the networking opportunities provided to computer science teachers. Both noted that attending Planning Days and connecting with other computer science teachers in the field has provided them with new ideas and resources. Wells adds, "I do a lot of reaching out to other computer science teachers and ask questions," and she credited some of her instructional successes to ideas that were originated during the Planning Days meetings.

Teachers interpret the meaning of a change initiative through the messages they receive from leaders and other stakeholders about how to engage with it (Allen, 2023; Lowell, 2024). They also make sense of messages through interactions with colleagues, where discussions further shape their understanding (Coburn 2002). However, when competing messages arise, they may result in uncertainty about implementation, which can lead to further ambiguity and confusion about how to carry out the initiative (Allen & Penuel, 2015). Participants in this study described receiving numerous and, at times, conflicting messages related to this change initiative.

Several participants shared messages of support through the words or actions of administrators, colleagues, and district leaders. Several participants indicated they received support from the district and administrative levels. Both Stone and Chase discussed the support they have received from their technology specialists in the form of newsletters containing resources and ideas. Wells, Locke, and Kerr explained that they receive the most support from their district and their building leadership (Table 9).

Table 9

Messages Received – Support

<i>Participant</i>	<i>District level</i>	<i>Building Administrator level</i>
<i>Kerr</i>	<i>“The computer science coordinator helped me understand the integration process.”</i>	<i>“I have an administrator at my school who encourages me and continually reaches out to help.”</i>
<i>Locke</i>	<i>“The people in the district have been really supportive of me as a new computer science teacher.”</i>	<i>“My principal’s been super supportive.”</i>
<i>Wells</i>	<i>“The district is where I get most of my support, from going to Planning Days meetings.”</i>	<i>“Our principal worked to get the position funded again.”</i> <i>“Admin has been amazing.”</i>

Additionally, some messages conveyed a lack of support, as teachers reported receiving contradictory messages regarding district expectations. Larkson expressed uncertainty about the implementation process, noting that while she understood computer science must be implemented in her classroom, she had not received any communication from administrators. She reflected that, “Since it’s new, maybe it’s coming,” and added, “Maybe they’re waiting on the computer science department to do it, but someone needs to spearhead it so that information starts to get where it needs to be, with the teachers.” Similarly, Clarke shared her confusion about their *Integration Engineer Implementation Plan* created by her team during P2P training which had not been used. “We’ve not included one mention of the plan in the past several years,” she said. When she sought clarification, administrators told the team that they, “still had some time.”

Participants also interpreted messages as assigning a relative value to the implementation of computer science. Some saw it through a practical lens, suggesting that teachers and administrators were preoccupied with more immediate priorities. Wells explained, “It’s not from any lack of knowing or not wanting to know, but you’re kind of the low man on the totem pole with your needs, versus reading, versus math, versus rosters, versus behavior.” Clarke, who had been told there was more time to implement the initiative, added, “I think even in our administrators’ minds, it’s just not a priority at all.” Others framed this perceived lack of urgency around testing. Wilder stated, “It’s not a tested subject, to be honest, and that’s kind of the way the education world is right now. If it’s not a tested subject, I mean, even social studies and science are on the back burner.” Kerr echoed a similar sentiment, “I don’t think they’re [administrators] worried very much about computer science. I think there’s more of the urgency to push literacy and math, because that’s what is on the TCAP test.” Lovette described being

discouraged from teaching computer science due to the presence of a dedicated computer science teacher in her school, recalling responses such as, “Oh, they’ll get that in computer class,” and, “Well, we’ve got a lot to do. Why don’t we let the computer teacher do that?” Stone added that after the initial push surrounding the computer science initiative waned, it appeared to be deprioritized in favor of other initiatives in ELA, math, and science. Both computer science teachers and non-computer science teachers identified the district’s computer science coordinator as the one who is communicating with them most about trainings and standards, concluded that outside of that there does not appear to be a district-wide commitment to the initiative.

Internal factors also play an important role in teachers making sense of a change initiative. In this study, those factors included change attitudes, technology attitudes, and self-directed learning. As Fives & Buehl (2016), noted, “Teachers’ beliefs shape their practice.” Similarly, Clayback (2022) found that individual teacher beliefs and experiences are key predictors of how a change initiative is implemented. Taken together, these findings suggest that teacher beliefs exert a strong influence on the success of implementation efforts. Because Research Question 2 seeks to identify the factors that influence teacher sensemaking, it is essential to consider these internal factors alongside external influences.

Most of the participants in this study expressed a general positive attitude toward change initiatives. Clarke describes herself as wanting to grow professional by embracing new initiatives and not settling for the status quo. Hughes reflected on the constant change she has experienced throughout her teaching career:

“I think because I have changed so much, both by choice and by necessity, I just roll with the punches. I made the decision to change from music to computers and had to learn all that entails and computers are constantly changing. I think it makes it easier for us [computer science teachers] to adapt because not only are

we having to adapt to different curriculum, different teaching, but also different technology, constantly.”

Kerr explained that she enjoys change because it is fun and breaks up the monotony of doing the same thing repeatedly. Wilder shared a similar perspective, stating, “To me, it’s fun to do the new initiatives.” Both Lovette and Locke echoed this openness toward change. Locke acknowledged that education frequently engages with new initiatives, which can leave teachers somewhat jaded and make buy-in more challenging but, emphasized the ultimate goal is always to do what is best for students. Lovette likewise shared that she strives to remain open-minded and eager to learn new things. While she admitted that she agrees with some initiatives more than others, she believes the guiding principle should always be acting in the students’ best interest. In contrast, Chase was candid about finding change difficult, admitting that she does not like it. However, she added that if the change aligns with her interests and her “brain is already primed for it,” she feels more indifferent and able to adapt.

In addition to attitudes toward change, this study also examined attitudes toward technology in general and classroom contexts. Many expressed positive views, emphasizing technology’s significance in today’s society and its value as an instructional tool. Others were more neutral or cautious, expressing concerns about its impact. A few participants reflected a mixture of enthusiasm and hesitation, revealing the complexity of teacher’s attitudes toward integration.

The majority of participants, approximately 70%, held positive views, describing a genuine appreciation for computers, technology, or technology integration. Larkson acknowledged that she loved computer science and spent much of her time on the computer, while Lovette shared that she enjoys computers but view them more as a personal hobby. Two participants, Locke and Wilder, explained that their early interest in computers had led them

consider careers in computer-related fields such as coding or information technology. Others distinguished between their affinity for computers and their feelings toward technology more broadly. For instance, Chase clarified that she enjoys using computers, her enthusiasm does not extend to her phone, which she finds more challenging to navigate. Both Kerr and Hughes described how their comfort with computers evolved over time, Kerr reflected, “I just fell in love with technology. I didn’t mean to, but I did,” while Hughes added, “I developed a knowledge and passion for incorporating technology into class, regardless of what kind of class it was.”

Participants emphasized the development of 21st century skills as a foundation for their beliefs about integrating technology and computer science into the curriculum. Across their comments a common thread emerged: preparing students for the 21st century is essential because technology integration is embedded in the modern workforce. Participants viewed computer science not as a passing trend, but as a necessary component of equitable education, ensuring that all students are prepared for the world they will inhabit. They expressed a shared belief that society is moving in this direction and that schools must equip students for the next step. Locke summed it up best: “If a kid wants to be a mechanic – computer science. If a kid wants to be an engineer - computer science. It is everywhere.”

Computer science can be incorporated into the curriculum to engage students in problem-solving and customize learning (Webb et al., 2017). Several teachers echoed those sentiments discussing the use of computers as a tool in education, and by extension computer science. Wells impressed upon her students the importance of using their computers to innovate and enhance their education, instead of consuming media, that it is a tool for learning and problem-solving. Wilder explains using Minecraft to teach specific computer science concepts to students and creates projects in Minecraft for students to demonstrate their learning, in as a

multicurricular way as possible. Locke uses computers as a “tool to further your thinking” and “to show how when you’re solving real world problems, the computer can be an aspect of that.”

Some participants shared mixed views of technology. One participant reflected that due to COVID, teachers were “thrust” into computer science, particularly in terms of increased computer use, but she also acknowledged that incorporating computer science elements into instruction could be enjoyable. She was not alone in that perspective, another participant admitted that she has chosen not to have a Promethean board in her classroom but recognized that other technology tools could be engaging for students. While generally positive toward technology, Locke cautioned that the primary focus should remain on the learner, emphasizing the importance of celebrating students’ learning above all else.

Many districts rely on professional development programs to prepare teachers for implementing new initiatives (Vegas et al., 2022). Professional development is recognized as critical when teachers are expected to adopt a change initiative (El-Hamamsy et al., 2020). Consequently, it is often viewed as an external factor that shapes teachers’ perceptions of their preparedness to implement change. Effective professional development can also empower teachers by enabling them to identify and respond to their own professional needs (Mushayikwa & Lubben, 2009). Such growth-oriented learning, however, depends on teachers’ intrinsic motivation and willingness to learn.

The will to learn is defined as, “having the ambition to discover new practices, being open to experiences and other people, being pro-active, attribution of successes and mistakes to internal causes, question-asking after performance, undertaking action to learn, and recognition of learning processes and results.” (Van Eekelen et al., 2006, p. 408). It has become an important factor in predicting the likelihood that teachers will adopt a new initiative (Mushayikwa &

Lubben, 2009). Because the will to learn is rooted in ambition and other internal processes and emerges from teachers' own initiative, this aspect of professional development shifts from being an external factor to an internal one (Van Eekelen et al., 2006).

Mushayikwa & Lubben (2009) outlined several factors that drive a teacher's self-directed learning, including the desire to strengthen subject content knowledge and to develop more practical knowledge and skills for a content area. Several participants identified as self-directed learners, having engaged in professional development to address their own needs. Larkson is a self-directed learner actively seeking out webinars and other opportunities to grow as a professional, explaining, "I like to take it upon myself to be more self-directed, and do my own research on different new initiatives that may be out there." Lovette describes herself needing to feel very prepared for a lesson and conducting research on new concepts in order to absorb it. Locke describes his self-directed learning as, "good old-fashioned searching around" to learn things on his own. He has even engaged in learning to code by taking a Python course.

To meet the learning needs of her students, who range in age from 5 to 22, Hughes must be able to decompose computer science concepts into developmentally appropriate and understandable ways. Holding a certificate in assistive technology, she is skilled at adapting curriculum to meet the diverse learning contexts and ensure accessibility for all students. She describes herself as "constantly seeking out more information on computer science for younger learners" noting that these materials are the most adaptable to her teaching. Hughes regularly participates in webinars to expand her understanding of computer science concepts and recently engaged with the PBS Kids learning webinars and a session focused on artificial intelligence.

As a new computer science teacher, Wells had never taught kindergarten before and described the experience as "very eye-opening." She initially felt confident in her preparation;

however, after implementing her first lessons, she realized that additional planning was needed. Wells reflected on spending evenings researching supplemental curriculum to improve the following day's instruction.

Summary of Research Question 2

Research Question 2 examined the relationship between internal and external influences and their role in participants' sensemaking. One overarching theme, *External and Internal Factors Shape Teachers' Preparedness* was identified, encompassing several important dimensions. Most of the external factors fell under the categories of *Professional Development* and *Messages Received*, both of which reflect the formal and informal social interactions central to sensemaking. Key external components influencing preparedness included the quality of professional learning opportunities and the clarity of communication. Internal factors also played a role, shaping perceptions of preparedness through their attitudes toward change initiatives and technology as well as their engagement in self-directed learning. Together, these internal and external influences illustrated how participants relied on their own frames of reference and interacted with external factors throughout the sensemaking process, contributing to their sense of preparedness.

Research Question 3

How do teachers perceive their preparedness for integrating computer science into the general elementary education curriculum?

This question explored the ways in which teachers experience and reflect on their sense of competence to teach computer science. It explored their feelings of self-awareness, and confidence, as well as any dispositions that influenced their overall sense of competence and engagement in learning opportunities designed to strengthen it. From this question, one

overarching theme emerged, *Building Self-Efficacy*, which captures how teachers perceive their preparedness to teach computer science.

Building Self-Efficacy

According to Bandura (1986), self-efficacy refers to an individual's belief in their capability to organize and execute the actions necessary to achieve the desired outcomes. In the context of teaching, it reflects a teacher's confidence in their capacity to bring about positive change and facilitate student learning (Grebing et al., 2023). Bandura identified four primary sources of self-efficacy: mastery experiences, vicarious experiences, social persuasion, and emotional arousal (Rich et al., 2017; Ryan & Hendry, 2023; Zhou et al., 2022). Participants in this study described experiences that align with these sources, often with areas of overlap among them (Table 10).

Table 10

Building Self-Efficacy Sources

<i>Bandura's Source of Self-Efficacy</i>	<i>Primary Category</i>	<i>Secondary Category</i>	<i>Explanation</i>
<i>Mastery Experiences</i>	<i>Confidence</i>	<i>Self-Awareness</i>	<i>Successes and reflections build belief in ability</i>
<i>Vicarious Experiences</i>	<i>Training</i>	<i>Disposition/Pre-disposition</i>	<i>Seeing examples and learning from others; shaped by prior experience and innate interest</i>
<i>Social Persuasion</i>	<i>Self-Awareness</i>	<i>Training</i>	<i>Encouragement through professional development</i>
<i>Emotional Arousal</i>	<i>Disposition/Pre-disposition</i>	<i>Confidence</i>	<i>Emotional state influences confidence and willingness</i>

Participants identified mastery experiences through moments of success that strengthened their confidence and reflections deepened their belief in their ability to teach computer science.

Locke recounted a successful lesson connected to the ELA text, *Hero's Journey*. In this lesson,

he and his students analyzed the story's structure, selected key scenes, and programmed a Sphero robot to navigate to each scene in the correct order, drawing parallels between sequencing a story and sequencing an algorithm. Students even programmed the Sphero to display emotional reactions at specific points in the sequence. Reflecting on this experience, Locke expressed satisfaction with how he was able to integrate coding into the ELA curriculum and felt confident in his ability to replicate similar lessons. He also noted areas for improvement, demonstrating self-awareness and a commitment to strengthening his instructional practice.

Chase also discussed a coding lesson that she implemented with her students to introduce algorithmic thinking and foundational concepts from the Tennessee Computer Science Standards. In this activity, students used a virtual Bee Bot – a digital version of the physical Bee Bot robot that is available on the manufacturer's website – to navigate correctly through the sequence of letters that spell their names. They then recorded themselves using the computer's built-in camera to explain their process. Reflecting on the experience, Chase identified repetition as a key factor in building her confidence to teach computer science, noting, "Every time I do something I'm much more confident in it." She also shared that when students approached tasks in unexpected ways, she would ask them to "show me how you did that," demonstrating her openness to reciprocal learning and recognition that she can learn from her students as they learn from her.

Participants engaged with vicarious experiences in a variety of ways, including through professional training. Wilder credited district-sponsored computer science Planning Days with strengthening his sense of self-efficacy, noting that feeling prepared is essential for effective instruction. He described two trainings that were particularly impactful. One early session featured a "make-and-take" lift-the-flap activity, designed to teach key computer science terms.

The activity supported computational thinking and vocabulary development by having teachers match terms with corresponding images and formal definitions before assembling them under the correct flaps. Wilder explained that this experience helped him clarify the distinction between *algorithm* and *decomposition*, two closely related concepts, and provided a practical classroom activity he could replicate with his students. In another training, Wilder participated in small group stations exploring computational thinking concepts. Each station included two activities focused on a specific concept, followed by group reflection on how these ideas could be integrated across the curriculum.

Wilder also described feeling particularly confident due to his affinity for computer science, which he attributed to his upbringing. He was already familiar with software such as Minecraft, which he now integrates into his teaching practice. Before becoming a computer science teacher, Wilder taught first-grade. He expressed confidence in incorporating ELA concepts into his computer science classroom, drawing on his prior teaching experience at that grade level.

By contrast, Lovette expressed a lower sense of preparedness and self-efficacy regarding the integration of computer science into the curriculum. Although she holds a master's degree in instructional design and technology and demonstrated a clear disposition toward working with computers, she does not currently integrate computer science regularly into her instruction. Reflecting on her experience during the COVID-19 pandemic, Lovette recalled feeling "a great sense of success," noting, "I found ways to help kids grow and learn from this computer screen using all these applications, and I just became enthralled with it and felt I was good at it." Despite this enthusiasm and proficiency with technology, Lovette explained that she still needs additional training and modeling to feel confident integrating computer science concepts into her

ELA curriculum. She described this as one of the greatest supports she needs, sharing that she does not yet see a clear connection between ELA content and computer science. Lovette characterized her level of experience as “emerging” but added, “If we could find a way to integrate that in our ELA program, I think we’d be rocking and rolling.” While her pandemic teaching experiences bolstered her general confidence in technology, the practical challenges of aligning computer science with ELA instruction have tempered her confidence in implementation.

Social persuasion is another source of self-efficacy that contributed to participants’ sense of competence and preparedness for teaching computer science. Wells reported feeling confident most of the time but acknowledged that much of her growth has come through trial and error. She realistically identified her strengths in working with students in third-grade and up, while expressing lower confidence with the younger grade levels. Wells described collaborating with the kindergarten team as particularly challenging and noted that she did not always feel supported by that group. Early in the year, she accepted the team’s concerns as reasonable given her limited experience with kindergarten. However, after her initial lessons did not go as planned, she redesigned them, pushed back on some of the team’s assumptions, and began asking students what sound a letter made and then challenging them to locate the corresponding letter on the keyboard worksheet. Although the students had to search for it, they were able to identify the correct letter on the coloring sheet. Reflecting on this experience, Wells shared, “it gives you confidence because you know they can do it.”

Wells reported receiving encouragement from her principal and from about 95% of the teachers at her school. However she emphasized that “the district is where most of my support comes from.” She attended a few district-sponsored trainings, including the Planning Days

sessions, which she credited for providing the guidance and support necessary to teach computer science. Reflecting on their impact, she explained “If I hadn’t been coming to those meetings, I can honestly say that I would’ve been lost because of everything I’ve taken from the meetings and brought back to my school.” Although she began teaching computer science in the 2025-2026 school year, Wells noted a marked increase in her confidence over a short time. She acknowledged that it is not typically in her nature to share information or share resources proactively, but when talking with colleagues at another school about passwords, she shared what she learned from the previous May’s Planning Days, including detailed instructions and copies of supporting material with them.

Clarke explained that she is working on integrating computer science vocabulary into her kindergarten instruction but admitted that she struggles to understand how to successfully integrate it with the existing curriculum. She also noted that she has not received guidance from school administrators regarding how to implement the plan the team created at the P2P training. In terms of social persuasion, Clarke described receiving mixed messages, there was an expectation to create a plan, yet little follow-through or support for enacting it. She reflected, “I know that I have some responsibility, so I try to find ways to incorporate ideas of computer science,” and added, “I just know we should be doing something. That should happen. I was part of a plan, but it just didn’t actually happen on the ground here at my school.” This situation illustrates how indecision can still constitute a decision, the expectation remains, but the initiative is effectively placed on the back burner. Clarke has made some attempts at integration, occasionally finding opportunities to use computer science vocabulary, though such instances are infrequent. She acknowledged that her sense of preparedness is so limited that she would not know where to start or how to proceed with her students. Clarke expressed a desire for more

training to help her understand and practice integration before being expected to teach it. Despite these challenges, she demonstrated a solid understanding of what computer science entails, recognizing the distinction between technology integration and computer science. Reflecting on her teacher preparation program, she recalled being shown what specific subject content looked like in practice and believes her sense of competence would improve if she were first taught how to teach computer science.

When Kerr began teaching computer science nine years ago, the focus looked quite different. At that time, digital literacy skills dominated the curriculum rather than computer science concepts. Kerr shared, “I’ve always been a quick study when it comes to computers,” and credited the new Tennessee state standards with helping her realize the need to evolve her teaching practice. She described her involvement on the Computer Science Steering Committee as a natural fit because its work naturally aligns with how she teaches students to think critically and problem solve.

As she explored coding programs, Kerr reflected, “When I think about people who are computer scientists, I think about people who can write code like Python or Java. Because that is a foreign language to me, it always scared me. However, seeing how you can code with blocks made more sense to me.” Kerr also identified several experiences that have contributed to her confidence and willingness to continue learning. Building a network of computer science teachers has been instrumental, giving her a community to lean on and learn from. Her confidence has grown substantially through her classroom successes and her role in helping to train other teachers. She also engages in ongoing, self-directed professional learning by experimenting with the same digital platforms she introduces to her students. As she explained,

“Just seeing the successes built my confidence. Even if not everybody fully understands it, at least they are trying and that is what I want them to do.”

Stone credited her teacher preparation program with shaping her disposition toward new initiatives. Although her exposure to computer science concepts has been limited, she believes her traditional teacher training provided the foundational knowledge needed to adapt to change.

“I feel that people who went the traditional teacher training route have less of a learning curve. There is still plenty to learn, but we at least have some background knowledge about basic teaching, so when something new comes along, I can connect it to things I’ve done before. So it’s easier to implement new initiatives, especially since education cycles through ideas.”

Her graduate program further helped her expand her instructional toolbox and learn to collaborate with others around new teaching concepts. These experiences have strengthened her overall confidence in adapting to new initiatives.

However, when speaking specifically about computer science, Stone expressed uncertainty, “I don’t feel like I have enough of an understanding about what computer science is asking students and what tool to give students to teach it effectively.” She finds the guidance from the state and district unclear, leaving her uncertain about what computer science instruction should look like on a daily basis. As a result, Stone’s confidence in teaching computer science remains low, particularly because she does not see clear connections between computer science and the curriculum she teaches. She believes her confidence would increase if the district provided professional development demonstrating how computer science can be integrated into existing curriculum. Additionally, she emphasized the need for clearer parameters, noting that current messaging discourages changes to the established curriculum. Stone also suggested that the district create a shared digital space, such as Schoology or Sharepoint, where teachers can access and share computer science integration resources.

Each participant was asked to rate their sense of preparedness on a scale of one to five, with one representing the least prepared and five representing the most prepared. One participant, Hughes, found it difficult to assign a single numerical rating because her sense of preparedness differs across the grade levels she teaches. She explained that her confidence is very high when teaching students with severe disabilities at the Pre-K level and remains strong when teaching grades K-2. However, was less confident working with students beyond that range. Hughes noted that she feels prepared to teach her middle school students with disabilities, but not “typical” middle school or high school students. Consequently, her data was excluded from the overall preparedness rating results.

The overall results revealed an average preparedness rating of 2.8 (Table 11). However, when disaggregated by role, notable differences emerged. Computer science teachers rated themselves significantly higher than non-computer science teachers, with an average score of 3.6 compared to 2.5 for non-computer science teachers, a difference of more than a full point. Those computer science teachers with the endorsement also rated themselves higher than those non-computer science teachers with the endorsement. Wells was excluded from these comparisons because she is a computer science teacher without the endorsement, which was not part of the criteria for this comparison. Given the participants’ reported experiences and their teaching roles it is not surprising that computer science teachers would have a greater sense of preparedness.

Finally, differences between veteran and the novice computer science teachers were modest, varying only by two-tenths of a point. This slight gap likely reflects the advantages of experience, as veteran teachers possess a broader base of knowledge and richer set of resources to draw upon. Hughes captured this sentiment, stating, “The more resources and the more information you have, the more prepared you should feel.”

Table 11*Preparedness Rating Results*

<i>Teacher</i>	<i>Preparedness Rating</i>	<i>CS Teacher</i>	<i>Non-CS Teacher</i>	<i>CS & Endorsed</i>	<i>Non-CS & endorsed</i>	<i>Veteran CS Teacher</i>	<i>Novice CS Teacher</i>
<i>Wells</i>	3	3					3
<i>Larkson</i>	3		3		3		
<i>Clarke</i>	1		1				
<i>Wilder</i>	3.75	3.75		3.75			3.75
<i>Locke</i>	4	4		4			4
<i>Lovette</i>	1		1				
<i>Kerr</i>	3.8	3.8		3.8		3.8	
<i>Chase</i>	4		4		4		
<i>Stone</i>	2		2				
Totals	2.8	3.6	2.2	3.9	3.5	3.8	3.6

When comparing the ratings of computer science teachers to those of non-computer science teachers, there is one major difference that stands out - sustained ongoing professional development. Computer science teachers participate in quarterly trainings that include instructional modeling, integration with the ELA curriculum, and exploration of available resources. Through these opportunities they continually engage in the iterative process of sensemaking. Although these trainings are also available to non-computer science teachers, they are not attending them, and therefore their sensemaking is not being shaped in the same way.

Another distinction exists between the computer science teachers with an endorsement and those without one. The most notable difference between these groups is that the computer science teachers focus primarily on that single content area. While they may integrate with ELA or other subjects, their central focus remains on computer science instruction. In contrast, the two non-computer science teachers with an endorsement are both fifth-grade teachers who teach specific content. They do not have the opportunity to teach, reflect, and re-teach computer science content daily, as their instructional time is dedicated to another content area.

Finally, a modest difference appeared between veteran and novice computer science teachers regarding to their perceived preparedness. One might expect the veteran computer science teacher to rate higher since due to greater experience with the content area; however, this finding should be interpreted cautiously, as only one veteran computer science teacher's data was included in the analysis.

Summary of Research Question 3

This research question examined the theme of self-efficacy and the various ways in which teachers assess their own sense of competence. Self-awareness emerged as a key factor influencing participants' confidence in integrating computer science. Professional development was particularly significant in shaping teachers' sense of preparedness. Participants who engaged in computer science professional learning attributed their confidence to successfully applying what they had learned in practice. Conversely, those who reported lower levels of confidence also expressed a greater need for professional development that provided clear, classroom-based models of implementation. Participants' self-assessments revealed differences in perceived preparedness, with computer science teachers demonstrating higher levels of confidence than their non-computer science counterparts. These findings reflect the outcomes of sensemaking, the iterative process through which teachers develop the deep understanding necessary to implement change effectively.

Summary of Findings

Across all three research questions, the findings revealed that sensemaking of the computer science change initiative unfolds as a gradual, iterative process that involves developing foundational understandings, responding to external and internal influences, and building self-efficacy. Educators engaged with the process from multiple entry points based on

their prior experiences. More experienced educators began with a deeper understanding and refined their practice as they encountered new information about the change initiative, whereas less experienced teachers relied initially upon familiar approaches and gradually expanded their knowledge and confidence over time. These findings provide the foundation for interpreting how teachers make sense of change and lay the groundwork for interpretation and implications discussed further in Chapter 5.

Chapter V: Discussion and Conclusions

Introduction

Computer science education has gained significant attention over the last ten years, and its integration into the K-12 curriculum is becoming increasingly widespread (Grove & Pea, 2018; Code.org, CSTA, ECEP Alliance, 2022). In Tennessee, the implementation of new computer science standards in the elementary curriculum was mandated to begin in the 2024-2025 academic year (TN DOE, 2022). For successful implementation, districts needed to understand the scope of these changes and provide teachers with time and support to make sense of the initiative. Sensemaking offers a valuable framework for understanding how educators interpret and respond to change. It is an iterative process through which educators construct meaning in ambiguous contexts, such as the integration of computer science into the elementary general education curriculum (Klein, 2007; Mills et al., 2010). Engaging in sensemaking enables teachers to develop concrete understandings of abstract concepts by interpreting both the external and internal inputs which in turn significantly influence the success of implementation (Dervin et al., 2008; Mills et al., 2010; Pirolli, 2011; Ancona, 2012; Stollman et al., 2020).

The purpose of this qualitative study was to explore the perceptions and experiences of elementary general education teachers regarding their capacity to integrate computer science into their curriculum. The data collected came from semi-structured interviews with ten teachers each participating in an individual interview. The interviews sought to examine how teachers made sense of a new curricular initiative, specifically, the integration of computer science. This chapter presents the study's conclusions, discusses potential implications, offers recommendations for future research, and identifies the study's limitations

Summary of Findings

This study sought to explore how teachers understand the integration of computer science into the elementary general education curriculum, the factors that influence their perceptions, and the perceptions of their preparedness. Through in-depth interviews, the findings revealed that sensemaking is an iterative process that unfolds gradually over time. It involves creating clarity out of ambiguity through both formal and informal social interactions. Across all three research questions, the data converged to tell a consistent story: sensemaking occurred as educators engaged with the change initiative by reflecting on their prior understandings, participating in professional development, and drawing on successful experiences that strengthened their self-efficacy.

Research Question 1 explored how teachers understood and experienced computer science integration. The findings demonstrated that educators began the sensemaking process by developing foundational understandings based on their prior knowledge and experience with computers. They engaged with instructional guidance designed to introduce policy, standards, and the practical realities of implementation. Teachers make sense of new initiatives more successfully when districts help them connect abstract expectations to concrete classroom-based meaning.

Research Question 2 examined the relationship between external and internal influences and their role in participants' sensemaking. Participants identified that their sensemaking was shaped through both formal and informal social interactions. Their understanding reflected experiences with professional development, the messages they received from school and district leadership, and their attitudes toward change, technology, and self-directed learning. Teachers' sense of preparedness for computer science implementation can be increased by districts that

encourage collaboration among educators and recognize the unique perspectives and life experiences that influence their willingness to adapt.

Research Question 3 focused on how teachers experienced and reflected on their sense of competence in implementing computer science into their curriculum. Teachers evaluated their self-awareness and confidence regarding computer science integration. Those who reported a higher sense of competence described positive experiences with professional development that met their needs. In contrast, teachers who expressed lower confidence often cited a lack of guidance or insufficient professional development to help them understand integration effectively. Teachers understand and feel more confident integrating computer science when districts provide clear instructions connecting professional development to the curriculum teachers are using.

Together these findings illustrate the cyclical nature of sensemaking for the computer science integration initiative. Teachers interpret new initiatives through the lens of their prior experiences, they refine their understanding through social interactions, and internalize their learning in terms of self-efficacy and competence. This process of interpret, refine, and internalize aligns with existing research and provides a roadmap for understanding how sensemaking unfolds during an educational change. Rather than prescribing what districts should do, these findings highlight why it matters to create conditions that support teachers' ongoing meaning-making.

Discussion of Findings

Research Question 1

How do elementary teachers understand and experience computer science integration into their general education curriculum?

The findings from this study demonstrated that teachers made sense of new initiatives through the lens of prior knowledge or experience with computers, implementing new policies into existing frameworks, and discovering challenges that shape as the process of sensemaking unfolds. In education, sensemaking is triggered by new or ambiguous situations often created by legislative mandates. Three key themes emerged in response to this question: *Understanding Begins with Core Concepts, Instructional Guidance, and Practical Realities of Computer Science Implementation.*

Understanding Begins with Core Concepts was evident across all participants in this study. The definitions they provided shared common ideas, emphasizing the cognitive aspect of computer science with focus on thinking and problem-solving, and innovation, as well as its use as a tool for accomplishing tasks. Participants also grounded their understanding of computer science by relying on their prior experiences. Notably, veteran computer science teachers described their understanding as an evolution, from what was once primarily viewed as digital literacy skills to the problem-solving applications used today. In contrast, teachers who were newly responsible for integrating computer science and had limited experience tended to define it more concretely, focusing on the physical act of using a computer. As Chase reflected, “you have to start from somewhere,” illustrating that sensemaking begins with existing knowledge. Similarly, Kerr recalled that when she first began teaching computer science, her primary goal was to help students navigate a computer. This progression aligns with what is known about the role of schema in policy implementation, how educators use prior knowledge to process new information and deepen understanding (Spillane et al., 2002). It also reflects the sensemaking process described by Luttenberg et al. (2013), in which teachers interpret an innovation through their personal frame of reference. These findings support the broader research on teacher

sensemaking by emphasizing that the process provides an entry point that resonates with teachers' existing understandings of a change initiative (Weick et al., 2005). When districts intentionally meet teachers where they are, they create a foundation for connecting abstract concepts with concrete understandings needed for successful implementation.

The theme *Instructional Guidance* illustrated how participants needed assistance in navigating the new initiative through formal, ongoing training. Interpretation of new policies and standards cannot be left up to teachers to interpret in isolation. This is particularly salient given that teachers begin new initiatives from varying points of understanding, and insufficient knowledge can interfere with their ability to construct understanding (Spillane, 2002). Chase explained that the computer science standards she already understood were easy to implement but those that she has “not figured out yet” would require additional time.

Her comments underscored an important distinction, as a math teacher, she is deeply familiar and comfortable with math standards, but less so with computer science standards. Over time, her experience with math standards fostered both competence and confidence, while computer science remains less central to her professional identity. The same was true for Lovette, who, when asked how her understanding influenced how she viewed herself as a computer science teacher, revealed that she primarily identified as an EL teacher and a fourth-grade teacher, but not as a computer science teacher. It is understandable then, that developing deep understanding is not high on their list of priorities. Their perspectives align with Firestone's (1989) assertion that teachers may selectively attend to standards that align with their own interests and areas of expertise. These findings reinforced existing literature about teacher sensemaking by illustrating that understanding how sensemaking occurs offers crucial insight into the cognitive and contextual processes that shape teachers' engagement with change

initiatives (Blignaut, 2008). This viewpoint reinforces the view that teachers interpret new initiatives through prior knowledge, experiences, and professional identities, highlighting the importance of aligning implementation with existing frameworks for understanding.

The theme *Instructional Guidance* demonstrated the importance of providing teachers with sustained, structured support as they learn to navigate the computer science initiative. Yet even with guidance, teachers encountered the complex, day-to-day realities of bringing the initiative to life in their classrooms. As participants engaged with the new expectations, they described obstacles that complicated implementation and revealed how systemic constraints interacted with individual sensemaking. From these narratives, the theme *Practical Realities of Implementation* emerged.

Participants described a wide range of engagement with the initiative, from fully adopting the computer science standards, to teaching foundational digital skills, to introducing related vocabulary, to not implementing at all. Because teachers began from different entry points, their implementation varied accordingly, contributing to uneven implementation across the district.

Several factors were cited as barriers to successful implementation of computer science, with lack of buy-in and time emerging as the most significant. Among computer science designated teachers, lack of buy-in was often attributed to minimal support from colleagues. Wilder, Wells, and Hughes all observed that computer science, as a related arts subject, is not perceived as holding the same importance as core content areas, a perception they viewed as central to the buy-in issue. In contrast, Stone, Lovette, and Clarke described tensions between district guidance to adhere strictly to the ELA curriculum and the expectation to integrate computer science. They explained that they neither saw clear connections between computer science and ELA nor received guidance on where or how to modify the ELA curriculum to

incorporate it. This finding suggests that the initiative lacked sufficient clarity and flexibility, placing the burden of adaptation on teachers (Schmidt & Datnow, 2005). The interaction between the initiative's design and teachers' sensemaking underscores the need for greater support in helping teachers interpret and apply new initiatives within their unique contexts.

Implementation, like sensemaking, is a gradual and iterative process. Participants noted finding time to engage meaningfully with an initiative that was both new and, for some, not yet professionally relevant. Teachers must carve out time to integrate computer science into existing curricula, an especially difficult task when ELA and Math blocks are tightly prescribed. Even for computer science teachers, time constraints limited the depth of implementation. In most schools, computer science instruction occurs within the related arts rotation, typically ranging from once per week to once every six to ten days, depending on school size. As a result, students received only about twenty to thirty hours of computer science instruction per year, forcing teachers to prioritize which standards to address. For meaningful change to occur, teachers must be given the time and space to make sense of the initiative. The findings of this study affirmed research on the gradual and iterative nature of educational change, reinforcing that teachers require time and space to engage in the full process of sensemaking. Such time is essential because it allows teachers to interpret new expectations, refine their understanding through collaboration and practice, and internalize what implementation looks like in their own classrooms.

Research Question 2

What factors influence teachers' perceptions of their preparedness for integrating computer science into the general elementary education curriculum?

The participants demonstrated that sensemaking is shaped through social interactions, both formal and informal. These interactions were influenced by a combination of external and internal factors that affected how teachers interpreted and responded to the computer science initiative. One key theme emerged from this question: *External and Internal Factors Shape Teachers' Preparedness*.

External factors played a key role in participant's sensemaking because they provided the context, information, and social interactions teachers relied on to interpret the computer science integration initiative. Participants described professional development, networking, and messages they received from others as influential in shaping how they understood and enacted implementation. Most participants viewed professional development positively, noting that it clarified expectations and strengthened their confidence. Hughes credited her involvement with the Computer Science Steering Committee as having, "given me more knowledge; otherwise I probably would not have a whole lot of that," demonstrating how external experiences expand teachers' understanding and reinforce their ability. This highlights the power of the social component of sensemaking as both an individual and collective activity (Coburn, 2002).

Participants also placed a high value on the messages they received from leaders and other stakeholders because those communications signaled the priority assigned to the initiative. In particular, non-computer science teachers expressed uncertainty about how to proceed with implementation, citing mixed messages from administrators. This ambiguity slowed their engagement with the initiative. These findings align with broader research showing that when teachers receive unclear or competing messages, more ambiguity is injected into the sensemaking process and hinders action (Allen & Penuel, 2015). Overall, these findings

emphasize that external factors shape not only what teachers know, but also how they interpret, prioritize, and act on change initiatives (Mills et al., 2010; Allen, 2023).

Participants also engaged with internal factors as part of their sensemaking process, as these reflect teachers' beliefs and motivations, which shape how they interpret new expectations and determine the extent to which they engage with implementation efforts (Fives & Buehl, 2016; Clayback, 2022). Most participants demonstrated positive attitudes toward change and technology, recognizing the growing importance of computer science in everyday life. As Larkson stated, "You really want to embrace those computer science integrations because it is where we're headed and we want our students to be equipped with it." Such beliefs are powerful drivers of sensemaking because they influence teachers' interpretations, how they assign meaning, and the confidence with which they act on new initiatives (Ancona, 2012; Fives & Buehl, 2016).

Additionally, participants, reported engaging in self-directed learning related to the computer science integration initiative. Wells admitted, "I do a lot of research on my own," adding, "I would go home at night and dig deeper." This form of intrinsically motivated professional development exemplifies how internal factors fuel ongoing learning and reinforce teachers' belief in their capacity to implement change. These findings align with broader research showing that sensemaking, as a social process, draws on both external and internal sources of information to interpret new expectations and construct shared meaning (Ancona, 2012; Gawlik, 2015). Taken together, external and internal factors enable teachers to translate understanding into action, and contribute to a shared districtwide vision of the change initiative (Lowell et al., 2024)

Research Question 3

How do teachers perceive their preparedness for integrating computer science into the general elementary education curriculum?

The findings demonstrated participants' sense of self-efficacy to implement change. Their confidence and competence were shaped by self-awareness, dispositions, and engagement in learning opportunities. From this question, one overarching theme emerged: *Building Self-Efficacy*.

Participants' sense of self-efficacy was examined through Bandura's (1986) four sources of self-efficacy – mastery experiences, vicarious experiences, social persuasion, and emotional arousal. Each source reflected factors that shaped how teachers perceived their own capacity to engage with the computer science integration initiatives. This sense of self-efficacy is critical because it influences how teachers interpret new expectations, persist through challenges, and decide which practices to adopt in their classrooms.

Mastery experiences included successful implementation of computer science lessons and reflective practice. Positive outcomes from these lessons strengthened teachers' confidence and reinforced their belief that they could meet the expectations of the initiative. Teachers built their sense of self-efficacy with each successful mastery experience (Rich et al., 2017; Zhou et al., 2022).

Vicarious experiences emerged through professional learning and personal interest in computer science. Opportunities to learn from modeling, common during Planning Days, allowed teachers to envision success, while an intrinsic comfort with technology, as Wilder described, further lowered barriers to implementation.

Social persuasion operated through encouragement and professional support. Wells, for example, cited the collective backing of her school community as reinforcing her confidence to continue computer science integration.

Finally, emotional arousal reflected the confidence and optimism that accompany readiness for change. Stone connected her traditional teacher training with the belief that she could adapt successfully to new initiatives.

Together, these sources explain why self-efficacy matters for sensemaking: when teachers experience success, observe effective models, receive encouragement, and feel emotionally ready, they are more likely to interpret change initiatives positively and engage in sustained implementation. These findings reinforce that building self-efficacy provides the foundation for teachers to make sense of, and act upon, educational change.

Synthesis Across Research Questions

The findings from this study highlight the iterative, social nature of sensemaking during a change initiative. Each research question examined a different phase of this process: the interpretation of a new initiative (RQ1), the external and internal factors that refine perceptions of preparedness (RQ2), and the internal assessment of one's preparedness (RQ3). A consistent narrative emerged: teacher's enter the sensemaking process with varying levels of understanding that are shaped by external and internal influences, ultimately leading to an internal evaluation of their readiness to implement the change initiative.

For RQ1, participants demonstrated varying levels of understanding about computer science. For example, Hughes exhibited a thorough understanding of the policy, while Stone reported having little to no understanding. Findings from RQ2 revealed that social interactions play a pivotal role in shaping teachers' sense of preparedness. Teachers engaged with both

external and internal factors to deepen their understanding and readiness. RQ3 affirmed that successful experiences enhance self-efficacy, further strengthening teachers' confidence in adopting the change initiative.

These findings align with the literature on sensemaking and educational change, which emphasizes how teachers construct meaning as they seek to bring order to ambiguity and uncertainty (Weick, 1995; Spillane, 2002). This study extends that understanding by underscoring the importance of intentional supports that address uncertainty and foster teacher sensemaking. District leaders who deliberately provide opportunities for teachers to interpret change in relation to prior knowledge, refine understanding through collaborative interactions, and create reflection experiences to internalize successes to build self-efficacy are more likely to see initiatives implemented with clarity and fidelity. Such an approach holds promise not only for computer science integration but also for broader change efforts requiring teachers to adopt new and unfamiliar practices.

Connections Between Emerging Themes and Existing Literature

These findings affirm that sensemaking is an iterative and social process involving interpretation and construction of meaning that unfolds when ambiguous or unclear situations arise in education, requiring participants to rely upon external and internal factors to tie abstract and concrete ideas together with the intention of implementing new initiatives. These findings are supported by a robust body of educational leadership research. The visual model below illustrates the alignment of theme ideas with literature on sensemaking, educational change, and self-efficacy. The visual model is organized to represent the dynamic process and multi-dimensional aspect of sensemaking.

The organization reflects *Interpret* → *Refine* → *Internalize*, a model consistent with sensemaking theory. The Interpret phase is the cognitive beginning of sensemaking where individuals find their entry point to the process. Teachers anchor new information in prior knowledge (Weick, 1995; Spillane et al., 2002). In the Refine phase, understanding evolves through social interaction, collaboration, and professional learning (Firestone, 1989; Cohen & Ball, 1990; Weick, 1995; Spillane, 2002; Coburn, 2005; Fullan, 2016; Darling-Hammond, 2017;). Finally, in the Internalize phase, teachers embed understanding in the belief that they can successfully enact new initiatives, sustained through systemic structures that reinforce and validate that confidence (Bandura, 1986; Firestone, 1989; Fullan, 2016; Darling-Hammond, 2017). This progression illustrates how individual and collective sensemaking becomes sustainable change in a cohesive system. Figure 2 provides a visual understanding of *Interpret* → *Refine* → *Internalize* with their corresponding theories.

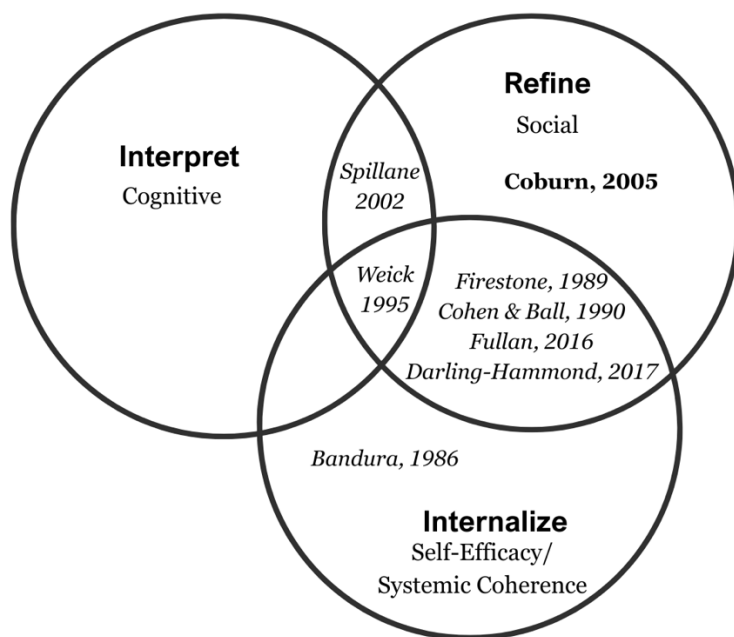


Figure 2: *Interpret* → *Refine* → *Internalize* & Theories

Recommendations for Practice

Computer science education has been mandated in many states across the United States; however, a persistent shortage of qualified computer science teachers poses a significant challenge to implementation, and Tennessee is no exception (Mason & Rich, 2019). Consequently, the integration of computer science into the general education curriculum, particularly in elementary schools, will primarily fall to in-service teachers. Unfortunately, most teacher preparation programs do not include computer science coursework or endorsements, leaving districts reliant on professional development initiatives to fill this gap (Vegas et al., 2022).

A well-prepared and knowledgeable teacher remains the cornerstone of effective instruction, yet identifying who will teach computer science highlights the complexity of this integration effort. Because teachers cannot effectively teach content they do not fully understand, districts must be deliberate in the design and delivery of professional learning for computer science integration. Recognizing the iterative nature of sensemaking, the following recommendations are offered to guide districts in supporting teachers as they interpret, refine, and internalize this change initiative.

Structured Interpretation

Understanding and implementing change is accomplished through the iterative process of sensemaking (Russell et al., 1993; Pirolli, 2011). This process takes time as teachers make sense of new information in relation to their existing beliefs and experiences (Spillane, 2002; El-Hamamsy et al., 2020). Initially, teachers will interpret a change initiative through their personal frame of reference, often focusing on aspects that align with their own area of experience (Firestone, 1989; Luttenberg et al., 2013). When districts introduce a new initiative, especially

one that requires engagement with new or unfamiliar content, they should intentionally design the early stages to support teachers' interpretation of the change. Structuring these opportunities as teacher-led promotes shared ownership and collective buy-in, both of which are strong predictors of successful implementation (Grebing et al., 2023).

For example, one participant in the study, Lovette, shared that she attended the P2P training, not because she understood its purpose but because her principal needed a team, and she “liked computers”. Her experience underscores the importance of guided sensemaking activities that explicitly communicate the “how” and the “why” of the initiative. Such activities can structure teachers' initial interpretations, helping them construct meaning and connect the initiative to their existing knowledge and classroom practice (Weick, 1995; Spillane et al., 2002).

Enhance Social Frameworks

Sensemaking is an inherently social process that relies on interactions with administrators and colleagues to develop a shared understanding of the reform's importance. (Coburn, 2001; Weick et al., 2005; Mills et al., 2010; Ancona, 2012; Luttenberg et al., 2013; Siciliano et al., 2017; Lowell et al., 2024). Social networks have a significant influence on how teachers engage with one another, often shaping and aligning individual beliefs with those of their peers (Siciliano et al., 2017). Kerr emphasized the value of collaboration, noting that her participation on the Computer Science Steering Committee was essential to her understanding because it gave her colleagues to “bounce ideas off of”. Her reflection affirmed the importance of establishing professional learning communities (PLCs) or “integration teams” where educators can collaboratively plan, share, and reflect.

In addition, non-computer science teachers identified a need for vicarious experiences that include modeling what integration should look like in the classroom. Providing opportunities

for peer observation or co-teaching enables teachers to learn from others' expertise and gain insights into effective instructional strategies for the initiative. Districts might also consider designating computer science coaches who can provide ongoing, job-embedded feedback and guidance. These strategies align with Bandura's (1986) concepts of social persuasion and vicarious experiences as key contributors to teacher self-efficacy and they reinforce the refine phase of sensemaking, in which understanding is deepened through social interaction (Weick, 1995).

Reflect for Self-Efficacy

Self-efficacy is a critical factor influencing teachers' motivation to implement a change (Lee & Min, 2017). It reflects one's belief in their capacity to affect positive change and achieve desired results (Grebing et al., 2013). Learning becomes internalized through reflection, as teachers evaluate their experiences and adjust their practice accordingly. For instance, Wells described teaching a lesson she believed she was well prepared for, only to later reflect on areas for improvement. She then engaged in self-directed learning to improve her lesson for the following day. This example illustrates how structured reflection tools, such as implementation journals, digital portfolios and teacher exemplars, can facilitate the internalization process and strengthen self-efficacy.

Districts may also consider developing self-assessment instruments for teachers to reflect on their own skills and comfort levels related to computer science standards. Additionally, confidence scales can be used to measure teachers' perceived self-efficacy following instructional delivery. Self-assessment and reflection promote internalization, reinforcing mastery experiences that cultivate stronger self-efficacy (Bandura, 1986).

Differentiate Professional Development

Effective professional development programs are essential because they will serve as the intermediary between implementation and classroom practice (Kong et al., 2020). Such programs play a critical role in supporting teachers as they adopt and implement new curricula (El-Hamamsy et al., 2020). Districts must consider both what educators need to know and how best to deliver training so that enables successful integration into existing instructional frameworks. Given teachers' diverse levels of experience, assuming a uniform perception of change can be counterproductive; differentiated support is therefore crucial for successful implementation (Hall, 1992).

Districts should design tiered professional development that acknowledges teachers' varying levels of computer science understanding. For example, entry-level sessions might focus on introducing non-computer science teachers to computational thinking and practical cross-curricular applications, while advanced training can provide computer science teachers with deeper content knowledge and integration strategies. Because teachers begin the sensemaking process from different starting points, these differentiated approaches ensure equitable access to learning opportunities and foster engagement by meeting teachers where they are in both understanding and confidence.

Communicate Clearly

District stakeholders play a critical role in curricular adoption by providing teachers with clear expectations, direction, and ongoing support (Spillane et al., 2002; Porter et al., 2015). When communication is inconsistent or unclear, teachers may experience uncertainty about implementation, leading to confusion regarding how to carry out the initiative (Allen & Penuel, 2015). For example, Stone expressed uncertainty about what computer science instruction should

look like, how it connects to her existing curriculum, and how it should be implemented. Her experience illustrated the need for districts to ensure alignment in communication among district leaders, administrators, and other stakeholders, to avoid mixed messages.

To promote coherence, districts should develop a clear implementation roadmap outlining expectations, available supports, and success indicators. In some cases, teachers appeared to understand policy expectations, but did not receive consistent communication or reinforcement from their administrators. Providing administrators with targeted training can help them interpret and reinforce the goals of the computer science initiative. Reducing ambiguity supports clarity and coherence, essential conditions for effective sensemaking and sustained change (Firestone, 1989; Coburn, 2005).

Cultivate a Culture of Continuous Growth

The sensemaking process involves the continual reinterpretation of a change initiative as individuals and groups construct new meaning over time (Weick, 1995). To achieve the goals of such initiatives, it is important to build teachers' capacity to engage in tasks they have not previously been asked to do (Darling-Hammond, 2005). Success lies in shifting the collective mindset to embrace and sustain the new initiative (Fullan, 2016). One effective way to shift collective mindsets is to embed computer science in schoolwide professional learning. While these trainings originate at the building level, they signal a heightened sense of priority and collective responsibility. The approach conveys a clear message about the importance of the initiative while fostering ongoing professional growth that aligns with the *interpret* → *refine* → *internalize* model of sensemaking. Additionally, publicly recognizing teachers who take initiative in computer science integration as leaders or change agents can further cultivate a culture of innovation and encourage others to take professional risks. When innovation, risk-

taking, and continuous professional growth are intentionally reinforced, meaningful and sustainable change becomes possible (Fullan, 2016; Darling-Hammond, 2017).

Recommendations for Policy

While the intention of the computer science integration initiative was meant to complement existing instruction in the elementary general education classroom, it was often perceived as adding another content area to an already full curriculum. Furthermore, the initiative requires significant time for teachers to engage in sensemaking and develop a deep understanding of computer science concepts. Given the existence of a K-12 endorsement pathway, and the fact that both computer science and non-computer science teachers now hold the endorsement, the state should consider requiring a fully-endorsed computer science teacher as part of the related arts rotation. Currently, implementation across the district lacks consistency in both instructional time and content coverage. Requiring computer science as a related arts subject would promote more equitable implementation and ensure full adoption of the standards.

Recommendations for Future Research

This research can be extended to explore additional dimensions of sensemaking and computer science integration that were not explored in depth in this study. From the findings, several areas of interest emerged that warrant further investigation. The following five topics stem from ideas closely connected to this research and offer potential directions for future study.

Formal Teacher Networking Groups

Several participants in this study who served on the Computer Science Steering Committee demonstrated notably stronger self-efficacy and deeper understanding of computer science integration compared to those that did not participate in that committee. Although the committee has been established for less than two years, its members consistently attributed their

confidence and preparedness to their involvement in this collaborative group. Future research could explore whether other content specific professional networks, such as professional learning communities (PLCs), yield similar effects on teacher sensemaking during change initiatives? Additional inquiry might examine how vertical alignment efforts within these group influence sensemaking across grade bands? Studying this work at scale could provide valuable insights for district leaders seeking to strengthen teacher networks and cultivate educators as agents of change.

Sensemaking for Administrators

Future research could examine how administrators make sense of new initiatives, particularly when they are faced with multiple competing priorities. Participants in the present study described receiving mixed messages from their building administrators regarding both the implementation timeline and the perceived priority of the computer science initiative. This raises important questions about the root causes of these inconsistencies and the ways in which administrators themselves experience the implementation process. The computer science integration initiative followed closely after the first year of a new, scripted ELA curriculum, potentially creating capacity challenges for administrators responsible for leading both efforts. Exploring how administrators navigate these overlapping reforms could help districts identify effective strategies for preparing and supporting building level leaders as they guide teachers through future change initiatives.

Scripted Curriculum

Participants in this study reported challenges integrating computer science into a scripted curriculum that allows little to no flexibility and requires strict adherence to pacing and content fidelity. This rigidity poses a significant barrier to implementing new initiatives. Future research

could explore how teachers engage in sensemaking when attempting to integrate innovative content within such constrained instructional frameworks. Additionally, examining how district-level staff responsible for new initiatives collaborate with those overseeing existing curricula could reveal strategies for aligning innovation with established expectations. Findings from this line of inquiry could inform how scripted curricula might be adapted or supplemented to support educational innovation while maintaining instructional integrity.

Longitudinal Study

This study included participants that entered the sensemaking process at different stages of understanding computer science. Most the non-computer science teachers reported their only formal exposure to professional learning was through the initial P2P training, with the exception of two participants who later earned their computer science endorsement. In contrast, computer science teachers participated in quarterly Planning Days – sessions open to all teachers - which provided ongoing opportunities for collaboration and growth. Future research could follow this group of educators longitudinally to gain deeper insight into how their sensemaking evolves over time and the factors that influence its development. Additionally, examining the effects of requiring all participants to attend Planning Days could help determine how consistent, content-specific professional development shapes teachers' sensemaking and implementation of the computer science initiative.

Additional Participants

This study focused on a small group of educators within a single state where the computer science integration initiative was introduced. Including both computer science and non-computer science teachers provided valuable comparative insights; however, expanding the participant pool could yield broader understanding of the phenomenon. Future research could

include a larger more diverse sample of teachers across multiple districts, states, or even nationally. Such an expansion would allow for stronger comparisons between teacher groups and offer a more comprehensive understanding of how sensemaking occurs around computer science integration initiatives. The findings could also inform best practices for supporting teachers and sustaining implementation efforts across varied educational contexts.

Limitations of the Study

One limitation of this study related to participant selection. Although the participants offered valuable insights, the candidate pool was limited by parameters established by the district. Including a larger number of non-computer science teachers may have yielded additional perspectives on sensemaking and implementation. Another limitation was the relatively low engagement in the computer science initiative among non-computer science teachers. Several participants were unable to reflect on computer science lessons because they had not implemented them. Finally, administrators were not included in this study; their perspectives on sensemaking and policy implementation could have offered important insights. Collectively, these limitations suggest opportunities for deeper understanding of how computer science is implemented and interpreted across different roles within the school system.

Conclusion

This study explored how teachers made sense of the computer science integration initiative, drawing on their prior knowledge, the factors that influenced their perceptions of preparedness, and their own assessment of their readiness for implementation. Participants described turning first to their existing knowledge as a foundation for understanding the initiative, emphasizing the critical role of both external and internal factors in shaping their sensemaking. These influences informed their self-efficacy and perceptions of preparedness for

implementation. The findings reinforce existing research on the iterative nature of sensemaking, importance of targeted professional development, and the role of self-efficacy in change initiatives. Participants' reflections demonstrated the need for clear, cohesive communication about policy changes and ongoing support for professional learning and implementation.

The results of this study highlight the importance of meeting teachers at their entry point to sensemaking, offering professional development that accounts for the constraints of scripted curricula, providing networking opportunities, and fostering experiences that build self-efficacy. These findings reaffirm that sensemaking is an iterative process that unfolds over time and is strengthened through continued professional growth aligned with the *interpret* → *refine* → *internalize* model. The study also illustrates how sensemaking matters, underscoring the need for intentional structures that promote ongoing meaning-making and sustained understanding.

Future research should continue to investigate how sensemaking occurs around new initiatives and how these initiatives are incorporated into existing curricula, including scripted programs. As schools continue to face shortages of qualified, computer science-endorsed teachers, providing all teachers with opportunities to expand their knowledge, skills, and confidence will be increasingly critical if states continue to mandate computer science integration into the general education curriculum. District leaders can use these insights to design implementation plans that strengthen teacher understanding, build capacity, and support ongoing professional development.

References

- Alaijlan, H., Alebaikan, R. & Almassaad, A. (2023). Computational thinking in K-12 computer education: Appropriate pedagogy. *Technology, Pedagogy and Education*, 32(3), 337-349.
- Alano, J., Babb, D., Bell, J., Booker-Dwyer, T., DeLyser, L. A., Dooley, C. M., Franklin, D., Frost, D., Gruwell, M. A., Israel, M., Jones, V., Kick, R., Lageman, H., Lash, T., Lee, I., Lyman, C., Moix, D., O'Grady-Cunniff, D., Owen, A. A., ... Weintrop, D. (2016). *K-12 computer science framework*. K12CS. <https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf>
- Allen, C. D. (2023). Figuring out my part in all of this: Understanding ambiguity and uncertainty in shaping teacher learning within reform. *American Journal of Education*, 130(1), 119-149.
- Ames, M. G. (2018). Hackers, computers, and cooperation: A critical history of Logo and Constructionist learning. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW).
- Ancona, D. (2012). Sensemaking: Framing and acting in the unknown. *The handbook for teaching leadership*.
- Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioural change. *Psychological Review*, 84(2), 191-215.
- Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Prentice Hall.
- Barefoot Computing. (n.d.) *Computational thinking concepts and approaches*. Barefoot Computing. <https://barefootcomputing.org>. Retrieved November 19, 2023.
- Baş, G., & Şentürk, C. (2019). Teachers' voice: Teacher participation in the curriculum

- development process. i.e.: *Inquiry in Education*, 11(1), 1–31.
- Berry, M. (2013). *Computing in the national curriculum: A guide for primary teachers*. Computing at School.
- Bers, M. U. (2008). *Blocks to Robots: Learning with technology in the early childhood classroom*. Teachers College Press.
- Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*. Oxford University Press.
- Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.
- Bers, M. U. (2022). *Beyond coding: How children learn human values through programming*. The MIT Press.
- Beycioglu, K. & Kondakci, Y. (2021). Organizational change in schools. *ECNU Review of Education*, 4(4), 788-807.
- Bhattacharya, K. (2017). *Fundamentals of qualitative research: A practical guide*. Routledge.
- Biggsby, J. B., & Firestone, W. A. (2017). Why teachers participate in professional development: Lesson from a schoolwide teacher study group. *The New Educator*, 13(1), 72-93.
- Bjursten, E., Nilsson, T., & Gumaelius, L. (2023). Computer programming in primary schools: Swedish technology teachers' pedagogical strategies. *International Journal of Technology and Design Education*, 33, 1345-1368.
- Blandford, A., Faisal, S., & Attfield, S. (2014). Conceptual design for sense making. In W. Huang (Ed.) *Handbook of human centric visualization* (pp, 253-284). Springer.
- Blignaut, S, (2008). Teachers' sense-making and enactment of curriculum policy. *Journal of Education*, 23, 101-126.

- Boyd, D. J., Grossman, P. L., Lankford, H., Loeb, S., & Wyckoff, J. (2009). Teacher preparation and student achievement. *Educational Evaluation and Policy Analysis, 31*(4), 416-440.
- Brouwer, N. & Korthagen, F. (2005). Can teacher education make a difference? *American Educational Research Journal, 41*(1), 153-224.
- Bryk, A., Sebring, P. B., Allensworth, E., Luppescu, S., & Easton J. Q. (2010). *Organizing schools for improvement: Lessons from Chicago*. The University of Chicago Press.
- Buck, S. (2017). When Steve Jobs donated 9,000 Apples to California schools, it was a tax and marketing coup. *Medium: Timeline*.
- Cabrera, L. (2019). Teacher preconceptions of computational thinking: A systematic literature review. *Journal of Technology and Teacher Education, 27*(3), 305-333.
- Century, Jeanne & Ferris, Kaitlyn & Zuo, Huifang. (2020). Finding time for computer science in the elementary school day: a quasi-experimental study of a transdisciplinary problem-based learning approach. *International Journal of STEM Education, 7*.
- Chenoweth, K. (2022). *Districts that succeed: Breaking the correlation between race, poverty, and achievement*. Harvard Education Press.
- Clayback, K. A., Williford, A. P., Vitiello, V. E. (2023). Identifying teacher beliefs and experiences associated with curriculum implementation fidelity in early childhood education. *Prevention Science, 24*, 27 – 38.
- Coburn, C. E. (2005). The role of nonsystem actors in the relationship between policy and practice: The case of reading instruction in California. *Educational Evaluation and Policy Analysis, 27* (1), 23-52.
- Code.org. (n.d.). *Computational thinking*. Code.org. <https://code.org/>. Retrieved November 19, 2023.

- Code.org. (2023). *Our core values*. Code.org. <https://code.org/about/values>. Retrieved November 18, 2023.
- Code.org. (2023a). *Support K-12 Computer Science Education in Tennessee*. Code.org. <https://code.org/promote/tn>. Retrieved November 20, 2023.
- Code.org. (2024). *Code.org's Approach to Diversity and Equity in Computer Science*. Code.org. <https://code.org/diversity>. Retrieved January 20, 2025.
- Code.org (2024a). *Tennessee: 2023 State of Computer Science Education*. Code.org. <https://code.org/assets/advocacy/stateofcs/2023/Tennessee.pdf>. Retrieved February 4, 2025.
- Code.org, CSTA, & ECEP Alliance. (2022). *2022 State of Computer Science Education: Understanding Our National Imperative. Computer Science - FAQs*. (2022, July).
- Cohen, D. & Ball, D. L. (1999). *Instruction, capacity, and improvement (CPRE Research Report N. RR-43)*. University of Pennsylvania, Consortium for Policy Research in Education.
- Cohen, D. K. (1995). What is the system in systemic reform? *Educational Researcher*, 24(9), 11-31.
- Conde-Zhingre, L. E. (2023). K12 framework of computer science and its importance in teaching-learning process as early ages. *2023 18th Iberian Conference on Information Systems and Technologies (CISTI), Information Systems and Technologies (CISTI), 2023 19th Iberian Conference On*, 106.
- Consoli, T., Désiron, J., & Cattaneo, A. (2023). What is “technology integration” and how is it measured in K-12 education? A systematic review of survey instruments from 2010 to 2021. *Computers & Education*, 197, 1-19.
- Creswell, J. (2009). *Research design: Qualitative, quantitative, and mixed methods approaches*.

Sage Publications.

- Crotty, M. (1998). *The foundations of social research: Meaning and perspective in the research process*. Sage Publications.
- Dariotis, J. K., Bumbarger, B. K., Duncan, L. G., & Greenberg, M. T. (2008). How do implementation efforts relate to program adherence? Examining the role of organizational, implementer, and program factors. *Journal of Community Psychology*, 36(6), 744-760.
- Darling-Hammond, L. (2005). Policy and change: Getting beyond bureaucracy. *Extending Educational Change: International Handbook of Educational Change*, 362-387. Springer Netherlands.
- Davis, T. (2019). New computer science standards define K-12 technology education. *Wyoming Business Report*. www.wyomingbusinessreport.com.
- Delgado, A., Wardlow, L. McKnight, K., & O'Malley, K. (2015). Educational technology: A review of the integration, resources, and effectiveness of technology in K-12 classrooms. *Journal of Information Technology Education: Research*, 14, 397-416
- Dervin, B., Naumer, C. M., & Fisher, K. E. (2008, April). Sense-making: A methodological perspective. In *Sensemaking Workshop, CHI* (vol. 8, pp. 506-513)
- Dorotea, N., Piedade, J., & Pedro, A. (2021). Mapping K-12 computer science teacher's interest, self-confidence, and knowledge about the use of educational robotics to teach. *Education Sciences*, 11.
- Dufour, R. (2007). In praise of top-down leadership: What drives your school improvement efforts--evidence of best practice or the pursuit of universal buy-in? *School Administrator*, 64(10), 38.

- El-Hamamsy, L., Chessel-Lazzarotto, F., Bruno, B., Roy, D., Cahlikova, T., Chevalier, M., Parriau, G., Pellet, J.-P., Lanarès, J., Zufferey, J. D., & Mondada, F. (2020). Computer science and robotics integration model for primary school: evaluation of a large-scale in-service K-4 teacher-training program. *Education and Information Technologies*, 26, 2445-2475.
- Elmore, R. F. (1995). Structural reform and educational practice. *Educational Researcher*, 24(9), 23-26.
- Ertmer, P. A. & Newby, T. J. (2013). Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. *Performance Improvement Quarterly*, 6(4), 50-72.
- Fink, D., & Stoll, L. (2005). Educational change: Easier said than done. *Extending Educational Change: International Handbook of Educational Change*, 17-41. Springer Netherlands.
- Firestone, W. A. (1989). Using reform: Conceptualizing district initiative. *Educational Evaluation and Policy Analysis*, 11(2), 151-164.
- Fives, H., Buehl, M. M. (2016). Teachers' beliefs, in the context of policy reform. *Policy Insights from the Behavioral and Brain Sciences*, 3(1), 114-121.
- Fluck, Andrew & Webb, Mary & Cox, Margaret & Angeli, Charoula & Malyn-Smith, Joyce & Voogt, Joke & Zagami, Jason. (2016). Arguing for Computer Science in the School Curriculum. *Educational Technology & Society*, 19, 38-46.
- Friebronn-Yesharim, M. & Ben-Ari, M. (2018). Teaching computer science concepts through robotics to elementary school children. *International Journal of Computer Science Education in Schools*, 2(3), 3-31.

- Fullan, M., & Stiegelbauer, B. (1989). *Implementing educational change: What we know*. Ottawa: Education and Employment Division, Population and Human Resources Department, World Bank.
- Fullan, M. (2016). The elusive nature of whole system improvement in education. *Journal of Educational Change* 17, 539-544.
- Funke, A., Geldreich, K., & Hubwieser, P. (2016, November). *Primary school teachers' opinions about early computer science education*. Proceedings of the 16th Koli Calling International Conference on Computing Education Research - Koli Calling '16.
- Glazerman, S., Isenberg, E., Dolfin, S., Bleeker, M., Johnson, A., Grider, M., & Jacobus, M. (2010). Impacts of comprehensive teacher induction: Final results from a randomized controlled study. NCEE 2010-4027. *National Center for Education Evaluation and Regional Assistance*.
- Grebing, Eric M., Julie A. Edmunds, and Nina P. Arshavsky. (2023). The relationship between buy-in and implementation: Measuring teacher buy-in to a high school reform effort. *Evaluation and Program Planning*, 97. doi:10.1016/j.evalprogplan.2023.102224.
- Grover, S. (Ed.). (2020). *Computer science in K-12: An A-to-Z handbook on teaching programming*. Edfinity.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Hall, G.E. (1992). The local educational change process and policy implementation. *Journal of Research in Science Teaching*, 29(8), 877-904.
- Haney, J.J., Lumpe A. T., Czerniak, C. M., & Egan, V. (2002). From beliefs to actions: The beliefs and actions of teachers implementing change. *Journal of Science Teacher*

Education, 13(3), 171-187.

Hargreaves, A. (Ed.). (2007). *Extending educational change: International handbook of educational change (Vol. 2)*. Spring Science & Business Media.

Hestness, E., D. J., McGinnis, J. R., & Plane, J. (2018). Professional knowledge building within an elementary teacher professional development experience on computational thinking in science education. *Journal of Technology and Teacher Education*, 26(3), 411-435.

Hew, K. F., & Brush, T. (2007). Integrating technology into K-12 teaching and learning: Current knowledge gaps and recommendations for future research. *Education Tech Research Dev*, 55, 223-252.

Huang, W. & Looi, C. (2021). A critical review of literature on 'unplugged' pedagogies in K-12 computer science and computational thinking education. *Computer Science Education*, 31(1), 83-111.

Huang, J. & Russell, S. (2006). The digital divide and academic achievement. *The Emerald Library*, 24(2), 160-173.

Hutchison, A., Colwell, J., Gutierrez, K., Evmenova, A., Offutt, J., & Taylor, V. (2021). Designing a model of computer science professional development for elementary educators in inclusive classrooms. *Journal of Technology and Teacher Education*, 29(2), 165-193.

Inan, F. A. & Lowther, D. L. (2010). Laptops in the K-12 classroom: Exploring factors impacting instructional use. *Computers & Education*, 55, 937-944.

International Society for Technology Education. (2023). *Computational thinking (CT)*. ISTE. <https://iste.org/computational-thinking>. Retrieved November 19, 2023.

- International Society for Technology Education & Computer Science Teachers Association. (2011). *Operational definition of computational thinking*. ISTE.
- Jacob, S. R. & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer Science Integration*, 1(1), 1-21.
- K-12 Computer Science Framework. (2016). Retrieved from <http://www.k12cs.org>.
- Kandemir, C. M., Kalelioğlu, F., & Gülbahar, Y. (2020). Pedagogy of teaching introductory text-based programming in terms of computational thinking concepts and practices. *Computing Applications in Engineering Education*, 29(1), 29-45.
- Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinness, J. R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology*, 29(1), 174-188.
- Kim, C., Gleasman, C., Boz, T., Park, H., & Foutz, T. (2022). Learning to teach coding through argumentation. *Computers and Education Open*, 3, 100107.
- Klein, G., Phillips, J. K., Rall, E. L., Peluso, D. A. (2007). A data-frame theory of sensemaking. In *Expertise out of context*, 118-160. Psychology Press.
- Kong, S., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers and Education*, 151, 103872.
- Kwon, K., Ottenbreit-Leftwich, A. T., Brush, T. A., Jeon, M., & Yan, G. (2021). Integration of problem-based learning in elementary computer science education: Effects on computational thinking and attitudes. *Educational Technology Research and Development*, 69, 2761-2787.

- Lake, S. (2020, April 29). *Amazon donates \$3.9M to CodeVA to support computer science education*. Virginia Business. Retrieved June 28, 2023, from <https://www.virginiabusiness.com/article/amazon-donates-3-9m-to-codeva-to-support-computer-science-education/>
- Lee, S. W. & Min, S. (2017). Riding the implementation curve. *The Elementary School Journal*, 117(3), 372-395.
- Lentz, R. G. (2000). The evolution of the digital divide in the US a mayhem of competing metrics. *Info*, 2(4), 355-377.
- Lewis, F., Edmonds, J., & Fogg-Rogers, L. (2021). Engineering science education: the impact of a paired peer approach on subject knowledge confidence and self-efficacy levels of student teachers. *International Journal of Science Education*, 43(5), 798-822.
- Liu, F., Ritzhaupt, A. D., Dawson, K., & Barron, A. E. (2017). Explaining technology integration in K-12 classrooms: a multilevel analysis model. *Education Tech Research Dev*, 65, 795-813.
- Lodi, M. (2020). Introducing computational thinking in K-12 education: Historical, epistemological, pedagogical, cognitive, and affective aspects [Doctoral Dissertation, University of Bologna]. Google Scholar.
- Lodi, M. & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education*, 30, 883-908.
- Lowell, B. R., Fogelman, S. E., & McNeill, K. L. (2024). Organizational sensemaking during curriculum implementation: The dilemma of agency, role of collaboration, and importance of discipline-specific leadership. *Science Education*, 108, 1448-1473.
- Lowther, D. L., Inan, F. A., Strahl, J. D., & Ross, S. M. (2008). Does technology integration

- “work” when key barriers are removed? *Educational Media International*, 45(3), 195-213.
- Luttenberg, J., vanVee, K., & Imants, J. (2013). Looking for cohesion: the role of search for meaning in the interaction between teacher and reform. *Research Papers in Education*, 28(3), 289-308.
- Ma, J., Wang, K., Zhang, Y., Liu, J., Bin, H., Gao, H. (2022). The development of students’ computational thinking practices in AI course using the game-based learning: A case study. *Proceedings of 2022 International Symposium on Educational Technology, ISET 2022*, 273-277.
- Macy, M., & Wheeler, T. (2021). Positive behavioral interventions and supports: Factors that influence teacher buy-in. *International Journal of Educational Organization & Leadership*, 28(1), 17–33. <https://doi.org/10.18848/2329-1656/CGP/v28i01/17-33>
- Marzano, R. J. (2003). *What works in schools: Translating research into action*. ASCD.
- Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, 189(4), 780-824.
- Mertala, P. (2020). The pedagogy of multiliteracies as a code breaker: A suggestion for a transversal approach to computing education in basic education. *British Journal of Educational Technology*, 52, 2227-2241.
- Microsoft launches K-12 education home page on World Wide Web. (1995, June 20). *PR Newswire*, 620NY067.
- Mills, A. J. & Thurlow, A. (2010). Making sense of sensemaking: The critical sensemaking approach. *Qualitative Research in Organizations and Management: An International*

- Journal*, 5(2), 182-195.
- Morel, R. P., Coburn, C., Catterson, A.K., & Higgs, J. (2019). The multiple meanings of scale: Implications for researchers and practitioners. *Educational Researcher*, 48(6), 369-377.
- Muhammad, A. & Hollie, S. (2012). *The will to lead, the skill to teach: Transforming schools at every level*. Solution Tree Press.
- Mushayikwa, E. & Lubben, F. (2009). Self-directed professional development – Hope for teachers working in deprived environments? *Teaching and Teacher Education*, 25(2009), 375-382.
- Northrup, A. K., Burrows, A. C., & Slater, T. F. (2022). Identifying implementation challenges for a new computer science curriculum in rural western regions of the United States. *Problems of Education in the 21st Century*, 80(2), 353-370.
- Novak, E. & Khan, J. I. (2022). A research-practice partnership approach for co-designing a culturally responsive computer science curriculum for upper elementary students. *Tech Trends*, 66, 527-538. <https://doi.org/10.1007/s11528-022-00730-z>
- Oda, M., Noborimoto, Y., Horita, T. (2021). International trends in K-12 computer science curricula through comparative analysis: Implications for the primary curricula. *International Journal of Computer Science Education in Schools*, 4(4),
- Oldham, J. (2021). Computer science for all? *Education Next*, 21, 44-51
- Oliver, C. E., Hicks, H. R., & Iles-Brechak, K.D. (1993). Adventures in supercomputing: A K-12 program in computing and computational science. *Proceedings of IEEE Frontiers in Education Conference – FIE '93. Frontiers in Education Conference, 1993. Twenty-Third Annual Conference. 'Engineering Education: Renewing America's Technology'*, *Proceedings*, 694-697.

- Orhan, A. (2022). The relationship between critical thinking and problem solving: A meta-analysis with correlational studies. Bartın University. *Journal of Faculty of Education*, 11 (3), 589-601.
- Ozturk, Z., Dooley, C. M., & Welch, M. (2018). Finding the hook: Computer science education in elementary contexts. *Journal of Research on Technology in Education*, 50(2), 149-163.
- Papert, S. (1993). *Mindstorms: Children, computers, and powerful ideas* (2nd ed.). Basic Books.
- Patton, M. Q. (2002). *Qualitative Research & Evaluation Methods* (3rd ed.). Sage Publications.
- Pietarinen, J., Pyhältö, K., Soini, T. (2019). Shared sense-making in curriculum reform: Orchestrating the local curriculum work. *Scandinavian Journal of Educational Research*, 163(4), 491-505.
- Pirulli, P. & Russell, D. M. (2011). Introduction to this special issue on sensemaking. *Human-Computer Interaction*, 26(1 & 2), 1-8.
- Plus Company Updates. (2017, December 6). *Infosys foundation USA to host free computer science training for over 800 public school teachers in summer 2018*. Plus Media Solutions.
- Pongračić L., & Marinac A. M. (2021). Teachers' Attitudes on the Development of Critical Thinking by Experiential Learning. *Acta Paedagogica Vilnensia*, 47, 108-121.
- Purpura, D. J., Schmitt, S. A., & Ganley, Colleen M. (2017). Foundations of mathematics and literacy: The role of executive functioning components. *Journal of Experimental Child Psychology*, 153, 15-34.
- Rana, D. S., Dhondiyal, S. A., Dimri, S. C., Malik, P. (2022). Impact of computational thinking

- in engineering and K-12 education. *Proceedings of the 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), Inventive Research in Computing Applications (ICIRCA), 2022 4th International Conference On*, 697-701.
- Rapaport, W. (2017). What is computer science? *APA Newsletter on Philosophy and Computers*, 16(2), 70-84.
- Resnick, M. (2017). *Lifelong kindergarten*. The MIT Press.
- Rich, P. J., Jones, B. L., Belikov, O., Yoshikawa, E., & Perkins, M. (2017, January). Computing and engineering in elementary school: The effect of yearlong training on elementary teacher self-efficacy and beliefs about teaching computing and engineering. *International Journal of Computer Science Education in Schools*, 1(1).
- Ross, B. H. (2001). Psychology of natural concepts. In N. J. Smelser & P. B. Baltes (Eds.), *International Encyclopedia of the Social & Behavioral Sciences* (pp. 10380-10384). Pergamon.
- Ross, S. (2020). Technology infusion in K-12 classrooms: A retrospective look at three decades of challenges and advancements in research and practice. *Educational Tech Research Dev*, 68, 2003-2020.
- Ryan, M., & Hendry, G. (2023). Sources of teacher efficacy in teaching reading: Success, sharing, and support. *The Australian Journal of Language and Literacy*, 46, 1-14.
- Ryoo, J. J. (2019). Pedagogy that supports computer science for all. *ACM Transactions on Computing Education*, 19(4), 1-23.
- Santo, R., DeLyser, L. A., Ahn, J., Pellicone, A., Aguiar, J., & Wortel-London, S., (2019). Equity in the who, how and what of computer science education: K12 school district conceptualizations of equity in “CS for all” initiatives [Conference Publication], 1-8.

Research on Equity and Sustained Participation in Engineering, Computing and Technology (RESPECT)., Minneapolis, MN.

Schmoker, M. (2018). *Focus: Elevating the essentials to radically improve student learning*.

ASCD

Shaw, D. E., Becker, H. J., Bransford, J.D., Davidson, J., Hawkins, J., Malcom, S., Molina, M., Ride, S. K., Sharp, P., Tinker, R.F., Vest, C. Young, J., Allen, R., Bakia, M., Bryson, R., Chen, C.S., Costello, C. M., Deckel, G. M., Dial, M. R., Kealey, E. M., & Lehoczky, S. (1998). Report to the President on the use of technology to strengthen K-12 education in the United States: Findings related to research and evaluation. *Journal of Science Education and Technology*, 7(2), 115-126.

Shulman, L. (1986). Those who understand: Knowledge growth in teaching. *Educational Researcher*, 15(2), 4-14.

Siciliano, M. D., Moolenaar, N. M., Daly, A. J., Liou, Y. H. (2017). A cognitive perspective on policy implementation: Reform beliefs, sensemaking, and social networks. *Public Administration Review*, 77(6), 889-901.

Smith, G., & Workman Publishing. (2020). *Everything you need to ace computer science and coding in one big fat notebook: The complete middle school study guide (Big Fat Notebooks)*. Workman Publishing Company.

Soini, T., Pietarinen, J., & Pyhältö, K. (2017). Shared sense-making strategies in curriculum reform: District-level perspective. *Improving Schools*, 21(2), 111-126.

Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M.L., Minsky, M., Papert, A., & Silverman, B. (2020). History of Logo. *Proceedings of the ACM on Programming Languages*, 4(HOPL).

- Spillane, J. P., Reiser, B. J., Reiner, T. (2002). Policy implementation and cognition: Reframing and refocusing implementation research. *Review of Educational Research* 71(3), 387-431.
- Sprott, R. A. (2019). Factors that foster and deter advanced teacher's professional development. *Teaching and Teacher Education*, 77, 321-331.
- Stager, G. S. (2016). Seymour Papert (1928-2016). *Nature*, 537(7620), 308.
- Stollman, S. Meirink, J., Westenberg, M., Van Driel, J. (2022). Teachers' learning and sense-making processes in the context of an innovation: A two year follow-up study. *Professional Development in Education*, 48(5), 718-733.
- Swain, C., & Pearson, T. (2003). Educators and technology standards: Influencing the digital divide. *Journal of Research on Technology in Education*, 34(3), 326-335.
- Tennessee Department of Education & Tennessee STEM Innovation Network. (2023). Computer Science: Integration Guidebook.
- Tennessee Department of Education. (2022). Computer science: Tennessee K-12 computer science state standards. https://www.tn.gov/content/dam/tn/education/standards/compsci/Final_Computer_Science_Standards_Framework_for_Grades_K-12.pdf
- Tennessee Department of Education. (2018). *Digital readiness: K-8 computer science standards*. https://www.tn.gov/content/dam/tn/stateboardofeducation/documents/2018_sbe_meetings/june-27,-2018-sbe-meeting/7-27-18%20III%20I%20K8%20Computer%20Science%20Standards%20Clean%20Copy%202.pdf
- TN.gov. <https://www.tn.gov/content/dam/tn/education/2022-legislative-session/Computer-Science-FAQs.pdf>
- Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., Verno, A. (2003). *A Model*

- Curriculum for K-12 Computer science: Final report of the ACM K-12 task force curriculum committee.* Association for Computing Machinery.
- Turnbull, B. (2002). Teacher participation and buy-in: Implications for school reform initiatives. *Learning Environments Research*, 5, 235-252.
- United Press International (1971, February 4). Computers employed as teaching aids. *Reading Eagle*.
- Van Eekelen, I. M., Vermunt, J. D., Boshuizen, H. P. A. (2006). Exploring teachers' willingness to learn. *Teaching and Teacher Education*, 22(2006), 408-423.
- Vegas, E., Hansen, M., & Fowler, B. (2021). *Building skills for life: How to expand and improve computer science education around the world.* Brookings Institute.
- Waterman, K. P., Goldsmith, L., & Pasquale, M. (2020). Integrating computational thinking into elementary science curriculum: An examination of activities that support students; computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology*, 29(1), 53-63.
- Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Syslo, M. M. (2017). Computer science in K-12 school curricula of the 21st century: Why, what and when? *Education and Information Technologies*, 22, 445-468.
- Wegner, Rachel (2024, September 25). Tennessee schools: How much do teachers make in your district? *The Tennessean*.
- Weick, K. E. (1995). *Sensemaking In Organizations.* Sage Publications.
- Weick, K. E., Sutcliffe, K. M., & Obstfeld, D. (2005). Organizing the process of sensemaking. *Organizational Science*, 16(4), p. 409-421.
- Whatley, A. E., & Smith, N. L. (2023). Writing in the content areas: Changing teacher

- perceptions with professional development. *Reading Psychology*, 44(6), 710-730.
- Williams, H. (2021). *No fear coding: Computational thinking across the K-5 curriculum* (2nd ed.). International Society for Technology in Education.
- Willingham, D. T. (2021). *Why don't students like school?* Jossey-Bass.
- Wilson, Cameron. (2013). What's up next for code.org. *Computer*, 46(8), 95-97.
- Wilson, Cameron & Sudol, Leigh & Stephenson, Chris & Stehlik, Mark. (2010). *Running on empty: The failure to teach K-12 computer science in the digital age*. Association for Computing Machinery.
- Wing, J. (2006). Computational thinking. *Communication of the ACM*, 49(3), 33-35.
- Wohl, B., Beck, S., Blair, L. (2017). The future of the computer curriculum: How the computing curriculum instills values and subjectivity in young people. *International Journal of Computer Science Education in Schools*, 1(1), 1-20.
- Woo, K. & Falloon, G. (2023). The search for computer science concepts in coding animated narratives: Tensions and opportunities. *Journal of Educational Computing Research*, 61(7), 1335-1358.
- Yang, X., Zhang, J. (2020). Preliminary comparison of K-12 computing education in China and the United States. *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Teaching, Assessment, and Learning for Engineering (TALE), 2020 IEEE International Conference On*, 965-967.
- Zhou, X., Shu, L., Xu, Z., & Padrón, Y. (2023). The effect of professional development on in-service STEM teachers' self-efficacy: A meta-analysis of experimental studies. *International Journal of STEM Education*, 10.

Appendix A

Interview Protocol

Participant Name:

Date of Interview:

Time Interview Began:

Time Interview Concluded:

Participant Pseudonym:

Participant Information:

Interviewer (I):

Thank you for participating in this study. As a reminder, the purpose of this qualitative study is to explore the perceptions and experiences of elementary general education teachers regarding their capacity to integrate computer science into their curriculum. The study aims to investigate how teachers make sense of a new curricular initiative – in this case, computer science integration – what factors influence that sensemaking, and how sensemaking impacts their assessment of their own preparedness to carry out a change initiative. Your participation in this research study is voluntary and you are free to withdraw from this study at any time with no penalty. This interview is scheduled to last for an hour.

Do you mind if I record our conversation?

Integrating computer science into the elementary general education curriculum is a new initiative mandated by the state. To understand how teachers make sense of this change initiative, I am gathering data to explore elementary teachers' perceptions, experiences, and readiness regarding computer science integration.

Your responses and your identity will remain confidential.

You will also receive a summary of your interview for your review as well as a copy of the interview, should you wish to review it in full, to check for accuracy and correct any information.

Some questions may prompt you to reflect on past experiences. If any question causes you discomfort, frustration, or stress, you are free to skip it. You may end the interview at any time. Just tell me you want to stop.

Do you understand everything so far? Do you have any questions?

May we begin?

Participant (P): Participant Affirmation(s)

1. Can you describe your background in education?

Probing: *How do you think your background has shaped the way you approach new initiatives?*

Probing: *In what ways do you think your prior experience impacts how prepared you feel to integrate computer science?*

2. Describe your background in computer science.

Probing: *How has your previous exposure shaped your confidence in teaching computer science?*

Probing: *Have you ever taken any computer science classes, in college or before? Can you describe those?*

3. How do you define computer science?

Probing: *How did you decide upon that definition?*

Probing: *How does your definition influence how you integrate computer science into your curriculum?*

Probing: *Do you think your understanding of computer science impacts how prepared you feel to teach it?*

4. What is your understanding of the policy regarding integration of computer science into the curriculum and the TN state standards?

Probing: *How does this understanding affect your approach to teaching and your confidence in teaching computer science?*

5. What guidance or communication have you received or engaged with regarding computer science integration at your school?

Probing: *How did these communications influence your readiness to implement it?*

Probing: *Who helped you understand the integration process? AND*

Probing: *How have these communications influenced how you think about integrating computer science?*

6. Tell me about any computer science lessons that you created or implemented with your students.

Probing: *How did teaching these lessons affect your confidence and sense of preparedness?*

Probing: *Can you share specific resources that you used and how you made decisions about what to teach?*

7. Based on what you told me, describe your overall experience with integrating computer science into the curriculum?

Probing: *How has this experience increased, decreased, or not changed your sense of preparedness? Why?*

Probing: *How has this influenced how you see yourself as a teacher of computer science?*

8. Tell me about your preparation to integrate computer science into your curriculum, did you attend trainings to learn more about computer science?

Probing: *Did they change how prepared you felt?*

Probing: *How relevant and helpful were these trainings?*

Probing: *How did they shape your understanding of computer science integration?*

9. Tell me about some of the successes you have had trying to integrate computer science into your curriculum.

Probing: *How did these successes affect your confidence in teaching computer science?*

Probing: *How did these successes shape your perspective on computer science integration?*

10. What challenges have you encountered trying to integrate computer science into your curriculum and what were the biggest obstacles?

Probing: *How did these challenges affect your sense of readiness to teach computer science?*

11. How would you assess your current level of preparedness for teaching computer science concepts to your students? You are welcome to use a scale of 1 to 5, with 1 being not prepared and 5 being extremely prepared.

Probing: *What are the reasons behind your feelings and what led you to this perspective?*

Probing: *Has your level of preparedness changed over time?*

12. What types of additional training, if any, do you think would enhance your ability to teach computer science even more?

Probing: *What kind of supports beyond training would improve your sense of preparedness?*

Probing: *How would this training address gaps in your current understanding or skills?*

13. Is there anything else that you would like to share?

Interviewer (I):

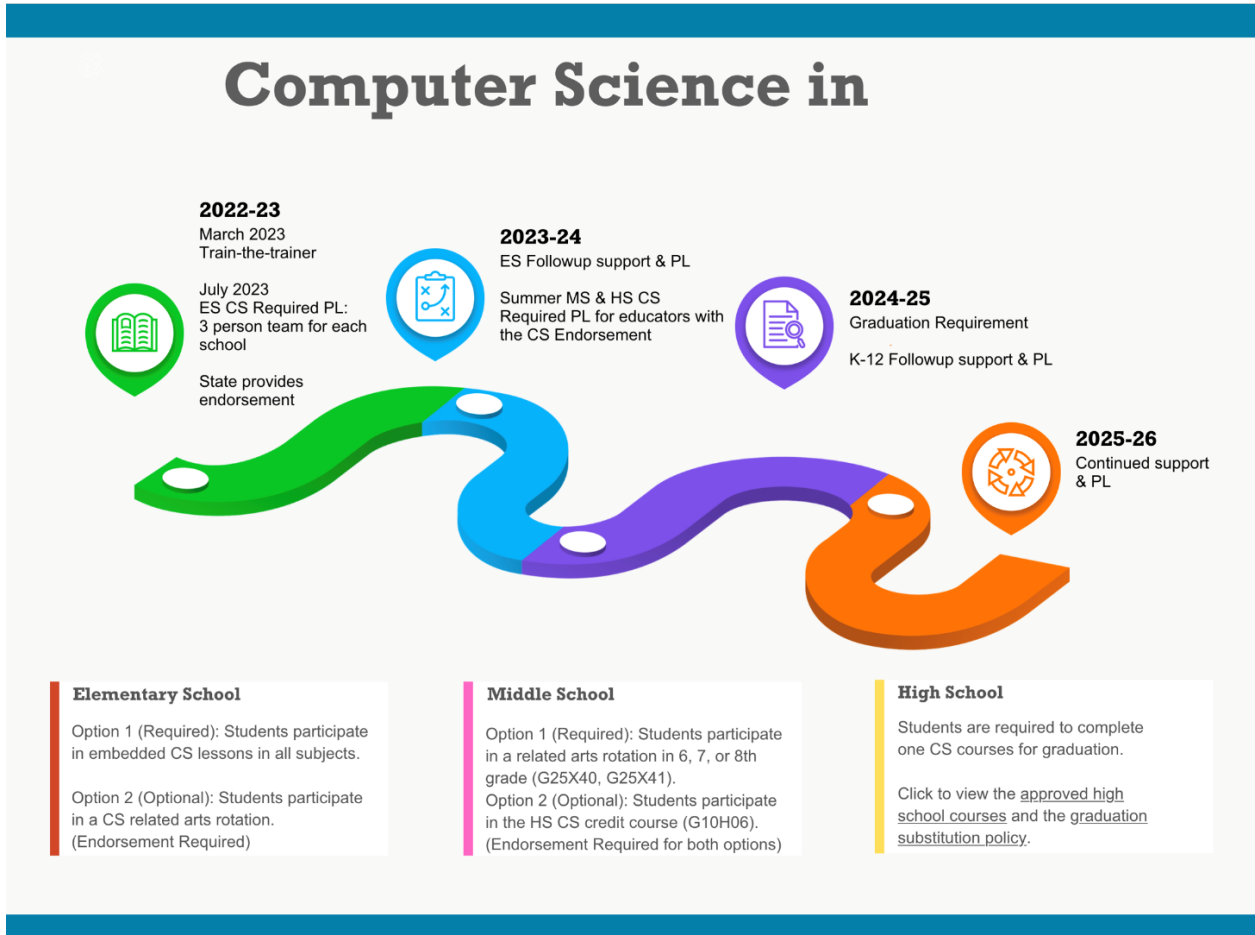
Those are all of the questions that I have. Do you have any for me?

Thank you again for your time and for sharing so candidly with me.

Appendix B

Artifact #1

District Roadmap



Appendix C

Artifact #2

TN State Policy (Tennessee Department of Education & Tennessee STEM Innovation Network, 2022)

Part A: Introduction to Tennessee Computer Science Law

Public Chapter #979 of the Public Acts of 2022

In 2022, Governor Bill Lee joined with the National Governors Association that includes fifty-five states and territories to sign a compact agreeing to take action and expand computer science opportunities for students in their states. This initiative and compact demonstrated growing momentum across the country and Tennessee for computer science education.¹

The Tennessee General Assembly in May 2022 passed by unanimous vote in both chambers [Chapter 979 of the Public Acts of 2022](#) (PC979), now codified in Tenn. Code Annotated Title 49, requiring specific actions be taken to ensure all students are fully prepared for the technological jobs of today and of the future. Concurrent with the passing of Chapter 979, the state of Tennessee joined 18 other states across the country allocating funding for computer science education.

Chapter 979 of the Public Acts of 2022 requires that high school students will receive one full school year of computer science education to satisfy graduation requirements by the 2024-2025 school year. Middle school students are to receive one course in computer science and elementary school students will receive grade-appropriate computer science education by integrating computer science standards into content.

These requirements will increase the availability of computer science courses and curriculum enabling more Tennessee students to benefit from enrolling in computer science education. The 2022 State of Computer Science Education Report indicates that 75.5% of Tennessee high school students attend a school that offers foundational computer science, but only 4.3% of students are enrolled in a computer science course. Of those students enrolled in a computer science course, 28.7% are female. With the passing of this law, Tennessee is now one of five states requiring students take a computer science course to graduate high school.¹

The Code.org Advocacy Coalition developed nine policy recommendations to make computer science a fundamental part of the state education system. These nine policies contribute to building and sustaining a comprehensive state policy framework that expands the teaching and learning of computer science.¹

¹ Code.org 2022 State of Computer Science Education: Understanding Our National Imperative