Fall 2023

The Need to Integrate Computer Science

Brittany Zimpfer
Northwestern College - Orange City

The Need to Integrate Computer Science

Brittany Zimpfer

Capstone Project: A School Improvement Plan

Northwestern College, Orange City, Iowa

Abstract

This school improvement plan outlines a detailed three-year strategy designed to integrate computer science into the K-5 curriculum. Emphasizing a comprehensive approach, the action plan employs a multi-tiered strategy combining a standalone curriculum with embedded activities. Drawing insights from successful educational practices and leveraging resources, the plan strategically aligns the curriculum with CSTA standards while fostering hands-on learning experiences at various grade levels. The timeline features foundational teacher training, curriculum integration, community engagement events, and consistent assessment processes. The plan aims to create an environment where both students and educators actively participate in the dynamic landscape of computer science education. By using a phased approach, this blueprint offers a comprehensive understanding of computer science concepts, equipping students for success in a technology-driven world. The plan acknowledges the importance of monitoring potential barriers and challenges to ensure effectiveness in the integration process.

Keywords: computer science, educational technology, elementary, integration

The Need for Integrating Computer Science K-5

Computer science, according to the State of Iowa, is defined as "[a]n understanding of how and why technologies work, exploring whether and how technology could solve real-life problems, investigating procedures, creating solutions, and learning about computing systems, programming, data, networks, and the effects on society and the individual. A study by Sun et al. (2022), provides strong evidence that overall, computer science education promotes the development of students' computational thinking in the K-12 setting while improving their creative and critical thinking skills. Computer Science is learning how to create new technologies, rather than simply using them" (Computer Science, n.d.). Iowa adopted computer science standards in 2018. According to the State of Iowa Computer Science Plan (2022), the new law will ensure that more students are ready for the computer science revolution reshaping the world far beyond schools. Iowa House File 2629 was signed into law in 2020 by Gov. Kim Reynolds requiring that all schools must provide high-quality computer science in all grade levels by July 2022 (Computer Science, 2022). The problem is that computer science needs to be integrated into all classrooms

The purpose of this school improvement plan is to address the need for improving the integration of computer science. The Iowa computer science mandate has left my school with the need to implement standards to meet the requirements of the law. At present, within the K-5 elementary school in rural Iowa Falls, Iowa, a single educator assumes the formidable responsibility of imparting computer science education to a student population of over 400. The teacher is also responsible for the STEAM curriculum.

Sources for this literature review were accessed via two primary sources: Google Scholar and the DeWitt Library at Northwestern College in Orange City, Iowa. Relevant scholarly journals published within the last decade were gathered and analyzed. The primary focus of the author's inquiry was on peer-reviewed research studies encompassing the realm of early elementary computer science education, aligning with the author's advocacy for kindergarten through fifth-grade levels. The research project's scope encompassed the spectrum from preschool to early elementary level, thereby enabling a comprehensive exploration of research pertinent to the integration of computer science education at a young age.

Integrating computer science standards into all academic disciplines is a necessity and an opportunity to empower our students with the knowledge and skills required to excel in a rapidly evolving, technology-driven world, helping to equip students with the tools they need to thrive in a rapidly changing environment. Such instruction would help contribute to innovation, and economic growth and address the complex changes of the 21st century. I am committed to promoting interdisciplinary learning, critical thinking, and creativity through the incorporation of computer science principles and standards across our educational curriculum. Research suggests that computer science standards integration has a positive impact on student learning more broadly. Arfé et al. (2019) revealed that students engaged in computer science activities demonstrated enhanced planning and inhibition skills within a one-month timeframe, contrasting to their counterparts who acquired these skills after engaging in standard activities for seven months.

The literature review that follows is organized into four main themes. It begins by discussing the legal and regulatory aspects of computer science standards in Iowa and then exploring best practices for implementing computer science education. These best practices include teaching fundamentals, hands-on learning, active engagement, and teacher professional development. Next, the literature review examines the implications of computer science in a global world including data-responsible use of technology. Finally, it examines the academic implications of integrating computer science into various disciplines. Implications include enhanced problem-solving skills, enhanced 21st-century skills, and the ability to personalize adaptive learning. Fessakis et al. (2013) discovered findings that indicated that the incorporation of computer science activities into other subjects can significantly enhance problem-solving skills in young learners.

Review of the Literature

This literature review explores various aspects of computer science education, covering four key themes. It begins with an examination of the legal and regulatory landscape of computer science standards in Iowa. The review then delves into best practices for implementing computer science education, emphasizing teaching fundamentals, hands-on learning, active engagement, and teacher professional development. Moving forward, it analyzes the global implications of computer science, including issues related to responsible technology use. Lastly, it investigates the academic impact of integrating computer science into different disciplines, focusing on improved problem-solving skills and the development of 21st-century competencies. Empirical evidence, such as the study by G. Fessakis (2013), will support the findings throughout this review.

Laws and Regulation

“Computer science has become a new basic skill in today’s economy and Iowa wants to ensure all of our K-12 students are offered high-quality computer science education that prepares them for personal and professional success in a digital world” (Computer Science, 2022). The State of Iowa believes that engaging students in concepts and practices of computer science through intentional progression will help prepare them to become inventive citizens who positively impact society (Computer Science, 2022). Governor Reynolds believes that investments in Iowa's PreK-12 public education system include a wide variety of work-based learning programs, STEM education, and computer science as early as preschool. She believes that by providing students with these experiences, we will have a future-ready workforce that is highly trained (Arend, 2022).

Former Governor Terry Branstad signed Senate File 274 into law in 2017 (Computer Science, 2022). In this legislation the goal was set forth to offer high-quality computer science instruction by July 2019. Governor Kim Reynolds signed House File 2629 into law in 2020, requiring for the first time that schools provide computer science instruction (Computer Science, 2022). The State of Iowa (2018) deemed necessary that elementary programming for grades 1 through 6 will encompass the following subjects: English-language arts, social studies, mathematics, science, health, human growth and development, physical education, traffic safety, music, and visual art. All students in publicly funded schools in Iowa are held to these subjects.

It is also stated that computer science instruction aligned with the standards outlined in rule 281—12.11(256) must be available starting from at least one grade level, beginning in the school year commencing on July 1, 2023 (Department of Education, 2020). The Computer Science Teacher Association (CSTA) Standards were adopted on July 1, 2018. The CSTA Standards Revision Task Force created these standards by merging conceptual statements and practices extracted from the framework. They also drew upon descriptive content from the framework to formulate examples and provide explanatory statements that complement the standards. The K-12 Computer Science Framework provides high-level guidance by grade band, while the standards provide detailed and measurable student performance expectations (Computer Science Teachers Association, 2017).

The CSTA standards encompass five main concepts; computing systems, networks and the internet, data analysis, algorithms, and programming, as well as the impacts of computing (Computer Science Teachers Association, 2017). Computer systems encompasses the study of computer hardware, software, and their interactions. It includes topics like computer architecture, operating systems, and computer networks. Networks and the internet equip students with the knowledge and skills necessary to navigate and comprehend the intricacies of modern computer networks and the global internet, which are essential in our interconnected world. (Computer Science Teachers Association, 2017). Data and analysis play a pivotal role in the realm of computer science education. This field revolves around the intricate handling of data and information, encompassing facets such as data representation, storage, retrieval, and manipulation.

Within this context, students acquire the proficiency to systematically structure, scrutinize, and derive informed conclusions from data sets Computer Science Teachers Association, 2017). Algorithms are step-by-step instructions for solving specific tasks or problems. Programming involves implementing these algorithms using a programming language. Understanding algorithms and programming is fundamental to computer science education. The impact of computing in today's world is crucial. In this area, students explore the ethical, social, and economic implications of computing and technology. They consider how computer science affects society, privacy, security, and the workforce. The CSTA concepts provide the structured framework for educators to design curriculum and instructional materials, ensuring that students gain a comprehensive understanding of computer science principles and skills necessary for today's modern world.

Key Terms

Defining key terms is helpful here, starting with *computational thinking* (CT). "Computational thinking refers to the thought processes involved in defining a problem and its solution so that the solution can be expertly carried out by a computer. We don't need computers to engage in computational thinking, but CT can leverage the power of computers to solve a problem" (Computer Science, 2021). Computational thinking skills relate to critical thinking and problem-solving across different subjects. They encompass the building of decomposition, abstraction, algorithmic thinking, and debugging and evaluation. (Computer Science, 2021). Another key term is *decomposition*. Decomposition is the process of breaking down a large problem into small parts, making a problem less overwhelming. *Abstraction* is the process of identifying common parts and recognizing patterns. It also includes sorting out the information we do not need "(Computer Science, 2021). Algorithmic thinking is designing a sequence of steps to accomplish a specific task. *Debugging* is the process of recognizing an error in an algorithm and testing and refining it until it completes the task (Computer Science, 2021). The last two vocabulary words defined are closely connected. *Unplugged computer science* is an innovative teaching approach that brings computer science concepts to life without the need for computers or digital technology. Instead, it engages students in interactive, physical activities and games that illustrate fundamental computer science principles. *Plugged* activities in computer science education refer to learning exercises and projects that are conducted with the active use of computers or digital devices. (Computer Science, 2021)

Key Terms:

- Computational—the thought processes involved in defining a problem and its solution so that the solution can be expertly carried out by a computer.

- Decomposition—the process of breaking down a complex problem into smaller parts that are more manageable, and helps us see problems as less overwhelming.
- Abstraction – identifying common features, recognizing patterns, and filtering out what we don't need.
- Algorithmic Thinking – designing a set of steps to accomplish a specific task.
- Debugging and Evaluation – testing and refining a potential solution, and ensuring it's the best fit for the problem.
- Unplugged activities- a teaching approach that brings computer science concepts to life without the need for computers or digital technology.
- Plugged Activities- refers to learning exercises and projects that are conducted with the active use of computers or digital devices.

Best Practices

In a time where computational thinking and digital literacy are crucial, knowing how to teach computer science effectively in educational settings is not just important, but vital. This review aims to explore research and emerging trends to uncover the best methods for promoting computer science as a core educational discipline. Additionally, it seeks to nurture a generation of students equipped with the skills and mindset essential for success in today's digital era.

del Olmo-Muñoz et al. (2020) conducted a study to assess the impact of incorporating "unplugged activities" on the development of computational thinking (CT) in early-year primary education students. Three research questions were posed: Firstly, which approach, either a mix of unplugged and plugged-in activities or solely plugged-in activities, fosters a more significant acquisition of CT skills during the initial years of Primary Education? Secondly, which approach, the mix of unplugged and plugged-in activities or solely plugged-in activities, yields better motivational outcomes when introducing CT to early Primary Education? Lastly, do significant differences exist in the effectiveness of these approaches concerning CT skills and motivation, taking into account the gender of the students? The study involved 84 second-year primary education students divided into four groups from different schools in the Castilla-La Mancha region of Spain (del Olmo-Muñoz et al., 2020). The instruction included two phases with a combination of code.org plugged and unplugged activities, emphasizing CT concepts suitable for the student's educational level, such as "directions," "sequences," and "loops."

The results indicated that the experimental group, which participated in unplugged activities in the initial phase of instruction, exhibited a more remarkable increase in CT skills. Furthermore, the experimental group demonstrated a significant advantage in the use of instruction that included unplugged activities, even when followed by plugged-in activities. Interestingly, the study suggested that motivation plays a more substantial role than gender in influencing outcomes. In conclusion, the study suggests that a mixed approach, combining both unplugged and plugged-in activities, is more effective for teaching CT in the early years of primary education compared to relying solely on plugged-in activities (del Olmo-Muñoz et al., 2020). The study's results strongly support the conclusion that the most effective approach for teaching computational thinking in early primary education involves a balanced combination of both plugged and unplugged activities.

In the pursuit of identifying best practices for teaching computer science, Fessakis (Fessakis et al., 2013) focused on the unique perspective of five- to six-year-old kindergarten children in a typical public semi-urban kindergarten school in Greece. The central inquiry revolved around the development of higher-order thinking and algorithmic problem-solving skills in these young learners. The study aimed to understand the feasibility and pedagogical value of specific computer programming activities within this age group (Fessakis et al., 2013).

Over 4-6 weeks, the study used activities from the National Library of Virtual Manipulatives as part of its intervention. This approach, rooted in qualitative and subjective educational research, sought to gain insights by interpreting the complex social phenomena through the participants' perspectives, emphasizing their views over creating theoretical models for formal experimentation. The findings highlighted several key considerations. First, the study explored whether kindergarten children were capable of problem-solving within programming environments, shedding light on the types of software-related problems they encountered. It also delved into the children's enjoyment of working with programming environments, their difficulties in solving problems, their problem-solving approaches, and the characteristics of their interactions throughout the process (Fessakis et al., 2013).

Additionally, the role and attitude of the teacher in facilitating these activities were examined, along with their opinions regarding the educational value of specific computer programming learning activities. The study underscored the importance of experiment-validated teaching and learning approaches, documented best practices, learning resources, curriculum standards, and professional development for teachers in the field of computer science education. It emphasized the need to cater to a wide age spectrum to offer developmentally appropriate learning opportunities consistently. Fessakis et al. (2013) study demonstrated that computer programming environments can support autonomous or guided open-ended explorations for young children, enabling active participation, critical thinking, and control over computers. Such environments were found to be conducive to creativity and intrinsic motivation in learning. The study also noted the availability of a variety of coding platforms that facilitate programming skill development (Fessakis et al., 2013). These insights contribute to the evolving best practices for teaching computer science to young learners.

In a study conducted by Hughes (Hughes et al., 2022), the focus was on how educational research informed the development of a virtual AI companion (AIC) to assist children in enhancing their social-emotional and communication skills. The project affiliated with Florida University was conducted in schools with a student population divided equally between neurotypical and students with disabilities. The study employed a mixed research approach and extended over two years, using RAISE Coding Programming and the multisensory teaching theory (Hughes et al., 2022). The findings of the study revealed that, despite the challenges faced with assistive robots, certain automated behaviors could be effectively provided to a diverse population with varying behaviors, social traits, and communication skills. The findings were based on individualized observations conducted during human-in-the-loop preliminary studies (Hughes et al., 2022). This study held some unique key components. One is students with autism spectrum disorder (ASD) learning to code with the assistance of an AI Companion (AIC) and a programming environment designed following Universal Design for Learning (UDL) principles (RAISE-UP). Students with ASD taught peers to code. Another interesting component was that professional development was provided to teachers to focus on cooperative learning in mathematics to enhance communication opportunities for students.

Importantly, the research demonstrated that participants, including students with disabilities, successfully learned to code and built upon their learning through explicitly taught steps. The study highlighted the significance of employing universal design for learning principles, affirming that all students can engage in coding. Furthermore, experimental results indicated that project-based learning integrated with robotic visual programs was significantly more effective in improving students' learning achievements compared to traditional teaching methods involving paper practice teaching materials (Hughes et al., 2022). The findings emphasize the importance of universal design for learning principles and project-based learning in enabling students, including those with disabilities, to successfully engage in coding and robotics education.

In Yang’s (Yang et al., 2023) multi-group experimental study, the primary objective was to assess the impact of a story-inspired approach in bridging the gap between programming and computational thinking (CT) among preschool children. The study was conducted with a diverse group of 108 children in a Chinese kindergarten serving ages three to six in Hong Kong. This comprehensive research initiative spanned multiple studies conducted over two to three years, using both plugged and reliable robots as well as debugging exercises (Yang et al., 2023). These children were exposed to one of three conditions: story-inspired robot programming (SIRP), story-inspired tablet programming (SITP), or unplugged CT education. This research fills a significant gap in the literature by investigating the use of culturally responsive teaching in the context of CT learning for preschoolers; the study builds on prior literature that recognizes storytelling as a potent and culturally responsive pedagogy.

Tsortanidou (Tsortanidou et al., 2019) underscores the critical importance of nurturing twenty-first-century skills, including creativity, critical thinking, and problem-solving, within the context of our globalized society. He emphasizes the shift from passive information consumption to active participation in a participatory culture, where students become creators and knowledge producers. Computational thinking (CT) is highlighted as a means to enhance creativity, involving the identification of computational elements in the world and the application of computer science tools to comprehend natural and artificial systems. Tsortanidou (2019) also underscores the collaborative and social nature of media arts practices and their role in cultivating intellectual and social skills.

He elaborates on the practical implementation of the described micro moments that aim to develop three crucial skills: computational thinking (CT), collaboration, and creativity. It emphasizes that these skills are inherent components of low-tech prototyping, unplugged computer science, and media arts practices, all of which are designed to enhance problem-solving, imagination, and social skills development. The proposed model seeks to unify these relationships and interrelationships, fostering a holistic development of the skills required by twenty-first-century students (Tsortanidou et al., 2019). The model engages learners in thinking, feeling, and doing, aligning with Papert's theory of constructionism. In the thinking realm, learners use their imagination to conceptualize ideas and apply creativity in their implementation, emphasizing the role of unplugged computer science, problem-solving, and computational thinking.

Like many of the studies listed above Tsortanidou (2019) emphasized the importance of developing three critical skills in computer science education: computational thinking (CT), collaboration, and creativity. These skills are seen as integral components of effective computer science instruction. He also concludes that best practices include principles like holistic skill development, scaffolding, multimodality, experiential learning, and the incorporation of arts and interdisciplinary elements (Tsortanidou et al., 2019).

In a systematic review of the literature done, Acevedo-Borrega et al. (2022) explores the current state of research on computational thinking in education. The study reviewed 149 articles. These articles were coded in a variety of keywords pertaining to computer science. The articles were also sorted into quartiles based on quality. The majority of the studies that were observed were studies done in the United States or Spain. The researchers concluded that adequate computational thinking (CT) in education requires a well-rounded understanding of CT, including its dimensions, pedagogy, and the effective use of technology. Achieving a balance among these components is crucial for achieving meaningful outcomes in computational thinking education. The need for teacher training has a high impact on success. According to Acevedo-Borrega (2022) requires pure knowledge of the computer science concepts as a whole, as well as an educator who is open to the process of learning and reflection.

A study by Mouza and colleagues (Mouza et al., 2022) involved a three-year summer institute for teachers focused on computer science education, with a subsequent follow-up classroom support program. The study included 94 teachers, primarily from a Mid-Atlantic state, of which 33% were male and 67% were female. The professional development program emphasized key principles, pedagogical strategies, broadening participation in computing activities, and more to enhance teachers' computer science knowledge and teaching skills. The summer institute featured activities and tracks on various computer science topics, such as algorithms, programming, and internet use. After three years, interviews were conducted with 28 teachers who participated in the program, eight of whom used all components of the professional development. Data was collected through surveys, interviews, and undergraduate reflective journals to explore how teachers applied their learning and the influence of professional development design features on teaching (Mouza et al., 2022).

Results of the study conclude that the program, including a summer institute, significantly improved teachers' knowledge of computer science content, pedagogy, and technology. It particularly helped enhance their understanding of fundamental computer science principles such as algorithms and programming (Mouza et al., 2022). Moreover, teachers gained pedagogical confidence, especially in areas like fostering student interest in computing and teaching computer science to underrepresented groups. The study found that teachers successfully applied their newfound knowledge in their classrooms, albeit with some adaptation to their specific contexts and student needs. Ongoing support, including collaboration with undergraduate students, proved crucial for effective computer science integration. However, the study's limitations include a self-selected group of teachers and the absence of direct classroom observations or student assessments. The study emphasizes the importance of structured professional development programs to help teachers gain knowledge of CS content, pedagogy, and technology. These professional development programs align with the best practice of providing teachers with ongoing and well-designed professional development opportunities to build their capacity for teaching CS effectively (Mouza et al., 2022).

Yune Tran (Tran, 2018) examined the impact of in-school computer science (CS) coding lessons on elementary-aged students in diverse suburban and rural schools. It found that the students' understanding of CS concepts significantly improved, including algorithms, loops, and debugging, following the intervention. The program incorporated various CS environments to engage students effectively, promoting equity by reaching underrepresented students in economically disadvantaged districts. The results indicated that early exposure to CS positively influenced students' attitudes, critical thinking, problem-solving skills, and interest in CS. Students also made connections between CS and other subjects, supporting interdisciplinary learning. The study emphasized the importance of teacher knowledge, pedagogical practices, and professional development in delivering meaningful CS education. The researchers suggested embedding soft skills like perseverance and collaboration into the curriculum. Future research should focus on assessing CT skills using revised instruments and investigating implementation factors and context-specific influences (Tran, 2018). The study corroborates the favorable outcomes observed in other research, thus reinforcing the positive implications for best practices.

In summary, incorporating a mix of unplugged (offline) and plugged-in (online) activities appears to be more effective in teaching computational thinking (CT) in early primary education compared to solely relying on online activities (del Olmo-Muñoz et al., 2020). Unplugged activities can lay a strong foundation for CT skills. Early exposure and exploration is important as kindergarten-aged children can engage in problem-solving and algorithmic thinking activities involving computer programming (Fessakis et al., 2013). Providing developmentally appropriate opportunities for young learners to explore computer science concepts can be highly beneficial. Using universal design principles in computer science education, especially for students with disabilities, can make coding and robotics accessible and effective (Hughes et al., 2022). Adaptations and accommodations can be made to support diverse learners. Using storytelling as a pedagogical approach, especially in culturally responsive ways, can bridge the gap between programming and CT in preschool children (Yang et al., 2023). Story-inspired approaches can engage young learners and make CT concepts more relatable. Providing scaffolding, guidance, and hands-on experiences in computer science activities can support students in developing problem-solving skills and CT (Acevedo-Borrega et al., 2022).

Experiential learning is a valuable approach in computer science education. Teacher training is essential for successful computer science education. Educators should have a deep understanding of CT concepts, pedagogy, and the ability to adapt to the learning process (Acevedo-Borrega et al., 2022). Ongoing professional development can enhance teaching effectiveness. These best practices emphasize the importance of a balanced, inclusive, and experiential approach to teaching computer science, starting from early education and extending throughout a student's academic journey (Mouza et al., 2022).

Implications of Computer Science on Global Society

In an increasingly interconnected and digitized world, the realm of computer science extends far beyond the boundaries of coding and programming. It permeates every facet of our lives, shaping not only how we work and communicate but also the very core of our global society. This section explores the implications of computer science in a global context. Focusing on data privacy, cybersecurity, and the imperative for responsible technology use becomes evident that computer science is not merely a field of study but a force that wields transformative power, bearing both opportunities and challenges that ripple across the globe.

Zhong & Zheng (Zhong & Zheng, 2023) discuss the pressing need to address the safety, privacy, and well-being of children in an increasingly digital world, where digital crime, trauma, and abuse have become amplified risks. The study acknowledges that although many children today are considered "digital natives" who seamlessly navigate both online and offline realms, they still require guidance to handle complex and evolving cyber threats effectively. Educators and researchers worldwide are urged to protect children while teaching them to use digital technologies responsibly and harness the opportunities they offer.

In the study by Zhong and Zheng (2023), a mixed-case research approach was employed, involving eight participants between 10 to 12 years. This study used a triangulation of data sources, including semi-structured interviews, concept mapping, and survey responses. The primary research inquiries addressed three key aspects: First, how do students at this age understand and conceptualize the concept of digital citizenship? Second, what are the dynamics of their interaction with a digital game after receiving direct instructional input? And third, what attitudes do these students hold toward digital citizenship? (Zhong & Zheng, 2023)

The study revealed that children understand that digital citizens should behave safely, legally, and ethically. They also show awareness of online risks, such as online fraud, addiction, privacy breaches, and cyberbullying. Zhong (2023) discovered that despite their understanding of certain digital citizen aspects, children's conceptualization of this concept was unclear or unorganized. I think the lack of understanding could be attributed to the limited exposure to lessons as well as age. The study did suggest that digital games such as Minecraft EDU had a positive impact on children's understanding of digital citizenship. This study could lead to future implications for game designers as they decide to responsibly expose children to a variety of immersive experiences.

The emergence of digital technologies has challenged traditional approaches to citizenship education due to the blurring of boundaries between online and offline worlds. Different conceptualizations of digital citizenship have arisen, leading to diverse approaches in digital citizenship education. Studies have explored teachers' perceptions, beliefs, and competencies in this context. Vajen et al. (2018) conducted qualitative interviews with nine teachers in Germany and eight teachers in Hong Kong, all of whom were active in the subject areas of civic or ethics education. The interviews focused on understanding teachers' beliefs about citizenship, the perceived impact of digitalization on citizenship, and the role of citizenship education in developing digital citizens. The participants were selected through purposive sampling to ensure a diverse sample of teachers with various teaching experiences. The data analysis involved qualitative content analysis, which categorized teachers' responses into four themes: 1. Citizenship 2. Digital Citizenship, 3. Competencies for Digital Citizenship, and 4.

Digital Citizenship Education. These themes covered teachers' definitions and opinions about citizenship, digital citizenship, the skills and competencies needed for digital citizenship, and the role of citizenship education. Additional categories were developed based on teachers' statements (Vajen et al., 2023).

Vajen et al. (2023) found that teachers in Germany viewed citizenship as membership in a community, emphasizing political participation and democratic norms. In contrast, teachers in Hong Kong associated citizenship with rights, responsibilities, and a sense of belonging. Both groups recognized the significance of digital citizenship, highlighting the changing landscape of information access and participation in the digital age, along with its associated challenges such as privacy concerns and disinformation. Digital citizenship education was seen as essential in preparing students for the complexities of digital citizenship, requiring competencies like technical skills, critical thinking, and ethical considerations. Challenges included the need for updated curricula, technical infrastructure, and teacher training, with suggestions for interdisciplinary approaches and curriculum adaptations. This study sheds light on the evolving concept of citizenship in the digital era and the crucial role of education in shaping responsible digital citizens (Vajen et al., 2023).

A study by Prasetiyo (Prasetiyo et al., 2023) assessed the digital citizenship competence of senior high school students in Central Java, Indonesia, using data collected during the COVID-19 pandemic through online questionnaires distributed via Google Forms. While my improvement plan primarily targets elementary students, I deemed it essential to incorporate data from high school students, as elementary students will eventually find themselves in similar circumstances. The research aimed to measure students' digital citizenship readiness based on the Nine Elements parameter, which includes protection, education, and respect. The survey incorporated a 5-point Likert Scale for responses, aligned with the Digital Citizenship Scale (DCS) criteria Prasetiyo et al. (2023). The respondents' demographic characteristics were analyzed using descriptive statistics and one-way ANOVA to explore the relationship between these factors and variables such as internet skills, internet attitudes, computer self-efficacy, and digital citizenship components.

Structural Equation Model (SEM) analysis was employed to assess the reliability and validity of the survey instrument. The analysis revealed that most respondents demonstrated high levels of internet attitude and skills, while variables related to digital citizenship fell into the medium to very high categories. (Prasetiyo et al., 2023). It's crucial to bear in mind that this study was conducted in a different country, and the results could potentially be influenced by cultural and societal factors.

In our increasingly interconnected world, computer science education extends beyond coding to shape responsible global citizenship (Zhong & Zheng, 2023). Studies show the need to guide children and students in navigating digital risks and understanding digital citizenship (Zhong & Zheng, 2023). As our concept of citizenship evolves, education plays a pivotal role in preparing individuals to participate responsibly in both physical and digital communities (Vajen et al., 2023). By integrating digital ethics into computer science education, we can foster a generation of global citizens who use technology for societal betterment while upholding ethical values, contributing to a more inclusive and equitable global society in the digital age (Prasetiyo et al., 2023).

Enhanced Problem Solving

Given the importance of problem-solving skills in computer science education today, one can draw inspiration from the results of a study by Ceylan Sen and colleagues (Sen et al., 2021).

The research investigates the computational thinking abilities of gifted students involved in STEM activities. These students engaged in hands-on experiences, particularly in robotics and 3D modeling, highlighting the significance of problem-solving in their learning journey. This section delves into similar studies focusing on the integration of coding and computational thinking in early education, emphasizing their role in shaping problem-solvers, promoting interdisciplinary learning, and enhancing student motivation. Through this exploration, we emphasize the vital role of computer science education in nurturing problem-solving skills, crucial in our ever-evolving digital world.

Sen et al. (2021) investigated the computational thinking skills of gifted and talented students participating in EDP-STEM (Engineering Design Process (Science, Technology, Engineering, and Mathematics) activities. The study involved seven gifted and talented students with varying talents in general ability, visual art, and music, who were enrolled in support education programs at the Science and Art Centre in Turkey (Sen et al., 2021). These students specialized in fields such as science, mathematics, technology, art, music, and literature; they also received targeted support education. The EDP-STEM activities consisted of hands-on, robotics, and 3D modeling tasks, with a focus on robotics activities as a sample. Prior to the robotics activity, students engaged in hands-on STEM activities to gain experience with the EDP. The robotics task involved solving Rubik's cubes using robots constructed from Lego Mindstorms and programming these robots. Following this, students engaged in 3D robot modeling and production using 3D printers, allowing them to identify real-world problems and create robots to solve them. Data was collected through STEM activity booklets, field notes, and video recordings. Content analysis was performed to investigate computational thinking skills, leading to the identification of various skills, including critical thinking, creativity, algorithmic thinking, problem-solving, communication, and cooperativity (Sen et al., 2021). The study ensured validity through data triangulation, expert opinions, and detailed descriptions of participants and context, while reliability was achieved through consistent coding and data verification.

The findings revealed that these students demonstrated algorithmic thinking, creativity, critical thinking, problem-solving, communication, and cooperativity skills within the realm of computational thinking. In the robotics activities, the students effectively employed algorithmic thinking by logically following problem-solving steps and developing creative solutions (Sen et al., 2021). They engaged in procedural coding, identified and fixed coding bugs, and successfully programmed robots to perform tasks like solving Rubik's cubes. The study also highlighted their ability to use coding as a language to equip robots with physical capabilities. Furthermore, students effectively employed critical thinking skills during EDP-STEM, demonstrated by questioning, reasoning, abstraction, association, making inferences, generalizations, providing justifications, and persuading others through explanations. Discussions played a crucial role in revealing these critical thinking skills. Additionally, the study showed that students excelled in making associations between STEM disciplines, other subjects, and real-life scenarios, thereby reinforcing their critical thinking and problem-solving abilities (Sen et al., 2021).

In Bers (Bers et al., 2019), 16 classes across three schools were involved in integrating coding, computational thinking, and robotics into early childhood education. Teachers underwent training on using a robotics kit and adapting it to their curricula. The study followed a single-group design, focusing on the entire group's learning outcomes, making comparisons with other groups unnecessary. Classroom sessions, lasting about 45 minutes, were conducted over two to three weeks, with teachers adapting the curriculum to suit their classrooms. Data collection involved structured observations of classroom dynamics, assessment of students' programming knowledge, and evaluation of positive behaviors using checklists. Additionally, teacher journals and interviews provided qualitative insights into the teaching experience. This comprehensive approach aimed to explore the effectiveness of integrating coding and computational thinking into early childhood education and its impact on both students and teachers.

In terms of coding and computational thinking skills, Bers (2019) focused on young children mastering fundamental concepts such as sequencing, repeats, conditionals, and debugging. Children showed high levels of understanding across these concepts, even as young as three years old, and demonstrated their ability to create complex programming sequences. Positive behaviors, as assessed through the PTD Checklist, highlighted strengths in communication, collaboration, community building, content creation, creativity, and choices of conduct, with communication and collaboration scoring particularly high. Teacher reflections indicated overall success in achieving teaching goals, though many teachers expressed the need for more extensive training and professional development (Bers et al., 2019).

The study highlights how coding and robotics can be integrated into various subjects, including language, mathematics, arts, and cultural studies. This interdisciplinary approach can enhance students' overall academic experiences and help them see the connections between different fields of study. Bers (2019) suggests that early exposure to coding, computational thinking, and robotics can lead to positive academic outcomes by enhancing students' problem-solving skills, interdisciplinary learning, positive behaviors, motivation, and engagement. Effective teacher training and a focus on cultural and social connections can further support these positive academic implications.

In Barbara Arfè (2011), the researchers investigated the impact of a short four-week coding training program using Code.org on the planning and response inhibition skills of first-grade children. They conducted a cluster-randomized controlled trial, with 80 first-grade children divided into an experimental group (coding training) and a control group (waiting list). The study assessed the children's coding abilities, planning skills, and response inhibition skills before the coding intervention, immediately after, and one month after the intervention. The coding training involved eight sessions over four weeks, with children working on coding problems using Code.org. The lessons gradually increased in difficulty, covering concepts like sequences, loops, and conditional instructions. The training aimed to maintain a problem-solving approach to coding tasks and was conducted individually in a computer lab by a trained post-graduate student (Arfé et al., 2011).

The results revealed immediate positive effects, with the experimental group outperforming the control group in planning and inhibition tasks after the coding training. These effects were sustained when the control group received the intervention (Arfé et al., 2011). Moreover, children exposed to coding became more efficient in problem-solving, as evidenced by decreased planning time and increased accuracy on coding tasks. Importantly, exposure to coding not only improved coding skills but also had transfer effects, enhancing planning and response inhibition skills. These findings highlight the potential of coding education to positively influence cognitive skills, even in young children aged 5–6 years (Arfé et al., 2011).

Egbert (Egbert et al., 2021) investigated the impact of coding education using the Ozobot platform in two 2nd-grade classrooms. Various data sources, including surveys, tests, interviews, observations, and student work, were collected to assess student engagement, perceptions, and knowledge gains during the coding lessons. The study acknowledged the challenges of conducting research in real-world educational settings and the short duration of the coding program, highlighting the importance of context-specific research in classrooms (Egbert et al., 2021).

The results of this study emphasize the importance of teachers' direct engagement with their students when learning new concepts like coding, as it enables teachers to better understand their students' capabilities and tailor instruction accordingly (Egbert et al., 2021). The study also employs a task engagement framework to understand classroom dynamics and learning processes. The research confirms that teachers benefit from hands-on experiences with their students when learning about new topics, allowing teachers to grasp challenges and reflect on their teaching practices effectively. The researchers suggest that providing teachers with orientation or workshops before implementing new instructional methods can be beneficial (Egbert et al., 2021).

The findings highlight the engaging nature of coding lessons with Ozobots for both teachers and students (Egbert et al., 2021). The study emphasizes the significance of providing students with a balance of structure and autonomy in their learning experiences, allowing them to test concepts thoroughly and develop problem-solving skills. The use of coding in multiple contexts and a problem-solving focus can help students transfer their knowledge effectively. The study emphasizes the value of hands-on learning experiences with coding and robotics, such as using Ozobots. This approach encourages students to actively engage with problems and challenges, fostering problem-solving skills through practical application. By teaching coding with a focus on problem-solving and connecting it to multiple contexts, students can learn to apply problem-solving skills beyond coding itself. This interdisciplinary approach helps students see the relevance of problem-solving in various aspects of their education.

In a five-year quantitative research study conducted in the Pacific Northwest, Lee (2015) assessed the impact of taking more credits in computer science courses at the secondary level on students' choices of STEM majors in postsecondary institutions in the U.S. The study involved high school students aged 15-18, and it used logistic regression analysis to assess the predictive power of various independent variables on the co-dependent variable of STEM major choices (Lee, 2015). The findings indicated that the quality of computer science education was as crucial as math and science education in motivating students to pursue STEM education and careers at the secondary level. Lee (2015) also highlighted the growing importance of computer and information technology occupations, projecting them to be among the fastest-growing job sectors through 2022. Notably, nearly three out of four STEM occupations were anticipated to require significant computing skills and knowledge (Lee, 2015).

Zeng (Zeng et al., 2023) addresses the global shift towards integrating computational thinking (CT) into early childhood education. Despite the potential of child-friendly programming tools, some students, particularly girls, may struggle to sustain their interest in programming. To address this, playful approaches, such as digital storytelling and culturally responsive teaching, are proposed to engage and motivate students effectively. These approaches blend traditional storytelling with digital technology, allowing children to explore programming concepts through narratives rooted in their cultural context (Zeng et al., 2023).

In this study, a multi-group experimental design with repeated measures was used to assess the impact of different interventions on children's computational thinking (CT) in early childhood classrooms in a Chinese kindergarten (Zeng et al., 2023). The interventions were randomly assigned to three groups of five-year-old children and lasted for nine weeks. The interventions included Story-Inspired Robot Programming (SIRP), Story-Inspired Tablet

Programming (SITP), and unplugged CT education (Zeng et al., 2023). Before the study, none of the classrooms had conducted any programming or CT education programs. The children were assessed using the TechCheck-K test, which measures various dimensions of CT. The results showed that both SIRP and SITP interventions significantly improved children's CT scores compared to the control group receiving unplugged CT education. Gender and family socioeconomic status did not moderate the effects of these interventions, indicating that both boys and girls from diverse socioeconomic backgrounds benefited equally from the story-inspired programming interventions (Zeng et al., 2023).

These findings highlight the effectiveness of culturally responsive pedagogy, using storytelling as an approach, in promoting CT skills in early childhood education. Additionally, the study suggests that integrating digital technologies like robots and tablets into programming education can enhance children's CT development, providing them with a more inclusive and sustainable learning experience (Zeng et al., 2023). CT skills acquired through story-inspired programming interventions transcend individual subjects. They promote interdisciplinary learning, where students can apply their problem-solving skills across various academic areas. Beyond academics, enhanced CT skills support lifelong learning and problem-solving in real-world contexts. Whether it's troubleshooting technical issues, making informed decisions, or navigating complex situations, individuals with strong CT skills are better equipped to succeed in their personal and professional lives.

Anastasia Misirli and Vassilis Komis (Misirli & Komis, 2023) highlight the natural problem-solving abilities of young children when exposed to coding activities. Their study explores the effectiveness of teaching preschool students (ages four to six) coding skills using the tangible robot Bee-Bot, particularly focusing on the practice of debugging. The research employs developmentally appropriate educational interventions and is the first to document debugging behaviors in young children concerning both syntactic and semantic knowledge. The study reveals that preschoolers exhibited debugging behaviors, even without prior experience or formal instruction, as they intuitively sought to make their programs correct and solve problems (Misirli & Komis, 2023). Two types of errors emerged: syntactic (related to programming structure) and semantic/logical (related to understanding the commands). The children demonstrated various debugging strategies, with many composing entirely new programs to correct errors. The study highlights the significance of developmentally appropriate programming tools and professional development for teachers to facilitate computational thinking and debugging skills in young children (Misirli & Komis, 2023).

The research shows that even very young children, aged 4-6, can engage in problem-solving behaviors when faced with errors in their code. The study also suggests that introducing coding and computational thinking at an early age can contribute to the development of problem-solving skills (Misirli & Komis, 2023). The findings revealed that children in the study exhibited problem-solving behaviors intuitively, without explicit instruction. Children in the study employed various problem-solving strategies, including composing entirely new programs to correct errors. This diversity in problem-solving approaches demonstrates the flexibility and creativity that can be fostered through coding activities (Misirli & Komis, 2023).

Chen-Chung Liu (Liu et al., 2011) considered the influence of simulation games on the computational problem-solving skills of 117 first-year university students in Taiwan who were novice programmers. These students were part of an "Introduction to Computer Sciences" course designed to develop their problem-solving abilities. The study introduced a simulation game called Train B&P, which allowed students to construct railway systems and control train behaviors within a 3D environment. The game incorporated physics simulations and required students to apply programming concepts. The experiment took place after 1.5 months of traditional programming lectures (Liu et al., 2011). Students were tasked with programming trains to complete specific tasks within the simulation game. Their programming activities and problem-solving processes were logged for analysis. To assess the impact of the simulation game, the study measured students' learning experiences; categorized them into flow, anxiety, and boredom states; and examined their motivations using the Motivated Strategies for Learning Questionnaire (Liu et al., 2011).

Liu (2011) aimed to reveal whether the simulation game positively influenced students' problem-solving skills and motivations compared to traditional lectures. The study also explored how different learning states affected students' problem-solving behaviors and strategies within the game. Findings from student feedback and activity logs indicated that the simulation game facilitated a flow experience, motivating students to employ trial-and-error, learning-by-example, and analytical reasoning strategies to develop problem-solving skills.

Students learning computational problem-solving with the simulation game were more likely to perceive a flow experience compared to traditional lectures. This result suggests that simulation games based on constructionism can improve problem-solving skills (Liu et al., 2011). Constructing a system retained students' focused attention on problem-solving and provided appropriate challenges conducive to a flow state. Intrinsic motivation also increased when students learned through constructivist approaches. The study revealed an association between students' learning experience states and problem-solving strategies within the simulation game. Students in a flow state were more inclined to use learning-by-example and analytical reasoning strategies, facilitating deep learning of computational problem-solving. The game's simulation of real-world physics and problem scenarios allowed students to apply their learning directly to solve practical problems (Liu et al., 2011). This hands-on experience is valuable for developing problem-solving skills that can be applied in real-life situations.

The importance of problem-solving skills in the context of computer science education is underscored by several studies that highlight their positive impact on learners across various age groups and educational settings. For instance, Sen et al.'s (2021) research involving gifted STEM students emphasized the significance of problem-solving in their learning journey, especially in hands-on experiences like robotics and 3D modeling. Similarly, Bers' study demonstrated that integrating coding and computational thinking into early childhood education can enhance students' problem-solving abilities and interdisciplinary learning (Bers et al., 2019). Furthermore, Arfé et al.'s (2011) research found that a short coding training program positively affected planning and response inhibition skills in first-grade children, emphasizing the cognitive benefits of coding education. Zeng et al.'s (2023) study on culturally responsive teaching and storytelling-based programming interventions in Chinese kindergartens revealed the effectiveness of these approaches in promoting computational thinking and problem-solving skills among young learners.

School Profile

Community Characteristics

All schools in the Iowa Falls School District are located in Iowa Falls, which has a population of 5,106 citizens (Explore Census, 2021). According to Iowa Census information, 89% of the population is white, while the other 11% is made up of African-American, Asian, and multiracial people. The median age of the average citizen is 41.6 years old. Accounting for the margin of error roughly 45 percent of the town's population are women while 40.3 percent are male (Explore Census, 2021). The community has a median household income of $53,313 (National Center, 2023). Iowa Falls has a diverse business landscape that encompasses various sectors, including agriculture, retail, healthcare, manufacturing, education, restaurants, and the service industry. Its strong agricultural tradition plays a significant role in the local economy, while a mix of retail shops, healthcare facilities, and educational institutions such as Ellsworth Community College serves the residents. Additionally, the town offers dining and hospitality services, as well as potential tourism-related businesses.

District Characteristics

Superintendent Dr. Tony Neumann leads Iowa Falls School District. The district serves preschool through twelfth grade. The total enrollment is 1,162 students. The district is prominently white at 86.7%, with a Hispanic population of 8.8%, and an African-American population of 1.1.% (Iowa School, 2022). The rest of the district is made up of multiple races, including Asian, Native American, and multi-racial students. The district is comprised of four schools. Pineview serves Preschool-1st grade, and Rock Run serves 2nd- 5th grade. Riverbend Middle School serves 6th-8th, and Iowa Falls-Alden High School serves 9th-12th grade for both Iowa Falls and Alden Communities. We remain separate districts with a grade-sharing agreement. Both Riverbend Middle School and Iowa Falls High School are labeled as Targeted status by the Iowa Department of Education (Iowa School, 2022). The district employs 95 full-time teachers with a student/teacher ratio of 13.41 (National Center, 2023).

School District Mission and Vision

The vision of the Iowa Falls School District is “to empower **all** to learn, lead and succeed at high levels both locally and globally. To achieve this, we are committed to excellence in communication, accountability, decision-making, effort, and teamwork. Excellence is defined as communicating effectively, being accountable for our words and actions, making sound decisions, giving maximum effort even when things are difficult, and working collaboratively to achieve goals (Iowa Falls Schools, 2022). The school's vision is the commit to excellence. Our school's focus is on five primary collective commitments that drive everything we do. These commitments revolve around communication, accountability, decision-making, effort, and teamwork for both staff and students.

1. **Communication:** This commitment emphasizes effective communication through various means, including verbal, digital, and nonverbal interactions. It encourages clear and purposeful messaging that leads to accurate information exchange. Staff and students commit to respectful and meaningful communication, embracing active listening and solution-focused conversations.
2. **Accountability:** Accountability is demonstrated through efficient time management, resource allocation, personal integrity, and self-monitoring. Both staff and students commit to taking responsibility for their actions, contributing to a positive environment, and recognizing the interconnectedness of their actions.
3. **Decision-Making:** This commitment emphasizes informed and reflective decision-making. Staff are committed to making decisions in line with the district's mission, vision, and values, utilizing evidence-based practices. Students commit to considering the impact of their decisions on themselves and others, making informed and responsible choices.
4. **Effort:** Effort involves giving one's best in all professional aspects, responding to situational needs, and embracing continuous growth. Both staff and students commit to

actively engaging in the learning process, persevering through challenges, learning from failures, and understanding their limitations to overcome them.

5. **Teamwork:** Teamwork requires collaboration, personal responsibility, and working together to achieve common goals. Staff commit to sharing strengths, collaborating, and monitoring progress in achieving team outcomes. Students commit to understanding their strengths and those of others, embracing their roles, and genuinely considering others' perspectives for collective success (Iowa Falls Schools, 2022).

Current Student Learning Goals

Student learning goals vary by building. Pineview Elementary has two large learning goals for the 2023-2024 school year. The first goal is by spring of 2024, the percentage of observed higher-order thinking skills will increase from an average of 12% per day to an average of 16% per day. The second goal is by spring 2024 is 100% of the classroom will engage in common instruction for 12-15 minutes per day to teach social and emotional competencies that support the collective commitments. (The baseline data for the second goal was 76.9%.) Rock Run Elementary's building goals are to implement the new literacy curriculum with intentionality and integrity. Its second goal is to prepare students for the future including next grade, next school, and beyond. Teachers are assessed by principals in random walk-throughs. When principals are doing these walk-throughs, they looking for teacher engagement, lower-order thinking skills, high-order thinking skills, and collaboration.

School Characteristics

Iowa Falls Schools are situated in the rural Northcentral region of Iowa, encompassing an area spanning approximately 30 square miles. This district comprises two elementary schools, namely Pineview Elementary and Rock Run Elementary. Pineview Elementary caters to a student body of 275 learners in preschool to first grade. The racial demographics of Pineview Elementary are 88.3% white, 6.8% Hispanic, 3.1% Multiracia, 1.2% African American, and 0.6% Asian (Iowa School, 2022). The gender distribution at Pineview Elementary reflects 54.3% male students and 45.7% female students (Iowa School, 2022). Pineview Elementary holds a position as a Title One school, with 37% of its students classified low socioeconomic status. In terms of language diversity, 3.7% of the student body are English language learners (ELL). Additionally, 11% of students have Individualized Education Plans (IEP) designed to address disabilities (Iowa School, 2022). Pineview and Rock Run both use a variety of state assessment tools. Both schools use Fast data. Rock Run also uses aMath, aReading, and ISASP.

Rock Run Elementary caters of 285 students in second to fifth grade. This cohort encompasses various racial backgrounds, with 86% identifying as white, 11.2% as Hispanic, 2.1% as multi-racial, 0.4% as African American, and 0.4% as Asian (Iowa School, 2022). The gender distribution within the student population is 55.8% male and 44.2% female. Notably, 15.4% of these students are eligible for an IEP due to specific disabilities, while an additional 5.6% were designated as ELL (Iowa School, 2022). Furthermore, Rock Run Elementary includes 41.1% of students as low socio-economic status. The student's academic performance is noteworthy, scoring 77.67/100 in English Language Arts, seven points above the state average of 70, and 76.21/100 in mathematics, more than 10 points above the state average (Iowa School, 2022).

Parent Involvement

In the Iowa Falls School District, parents stay informed through the PowerSchool platform, Google Classroom, and weekly school-specific newsletters. Parents have the option to engage in biannual parent-teacher conferences, providing them with the chance to discuss various matters with teachers. These discussions may encompass student behavior, academic performance, or any other school-related subjects. Additional forms of participation involve joining the Parent Teacher Organization (PTO), as well as securing positions on the school board, music booster club, wrestling booster club, athletic boosters, and coaching staff.

Current Computer Science Curriculum

Currently at both Pineview Elementary and Rock Run Elementary computer science is exclusively taught in STEAM (Science, Technology. Engineering. Art, and Math). Computer Science for All in San Francisco is used in kindergarten through second grade. The Computer Science for San Francisco curriculum is a comprehensive educational program designed to teach computer science and computational thinking to students in the San Francisco Unified School District and beyond. It encompasses various grade levels and focuses on developing students' problem-solving skills, creativity, and critical thinking abilities through coding and computational activities. The curriculum covers a wide range of topics, including algorithms, data structures, and computer programming, and it encourages students to engage in hands-on activities using various programming languages and tools. It aligns with national and state standards and aims to provide equitable access to computer science education for all students in San Francisco and abroad. This curriculum is available for free online (CS for SF, n.d.).

In third grade through fifth grade Computer Science First from Google (CS First) is the chosen curriculum. CS First from Google is an initiative aimed at introducing computer science to students in a fun and engaging way. It provides free resources, including instructional materials, coding activities, and online tools, to help educators teach computer science concepts to students in elementary and middle school. The program focuses on fostering creativity, problem-solving skills, and collaboration through the use of block-based coding languages. It offers a variety of themed projects, which allows students to explore different topics and create interactive animations, stories, and games. CS First aims to make computer science education more accessible to a broader audience, particularly targeting underrepresented groups, and encourages students to develop a passion for technology and coding. It is also aligned to state and national standards (CS First, 2022).

A variety of robots are in use throughout the STEAM program. The robots used in kindergarten are Bee-Bots, which are simple, durable robots designed for early childhood education. These robots are specifically created to teach young children the fundamentals of programming and problem-solving. Bee-Bots can be programmed with directional commands and are typically used for activities that enhance critical thinking and sequencing skills. For first grade, Indi is used. It is a programmable robot by Sphero, designed for K-2 students. It is a versatile educational tool that allows young learners to explore coding concepts through hands-on activities. Indi offers various programming options, making it a valuable addition to elementary classrooms.

Third grade uses Ozobot, which is a small, programmable robot designed to introduce young learners to coding and robotics. It operates on both physical and digital surfaces, allowing students to draw lines and patterns with markers to create code sequences. Fourth grade explores computer science concepts with the Finch robot. It was created by researchers at Carnegie Mellon University's CREAT Lab. It can be programmed using various coding languages. It is equipped with various sensors that allow students to engage in projects that involve real-world data collection and analysis. Fifth grade has been very enthusiastic about the use of drones in computer science curriculum. Tello EDU drones promote interdisciplinary learning by allowing students to explore connections between computer science, mathematics, physics, and engineering. Students can investigate concepts related to physics and aerodynamics as they program the drones to take off, land, and fly.

Integrating robotics into the computer science curriculum offers several compelling reasons for educators and students. First and foremost, it provides a hands-on, interactive approach to learning computer science concepts, making abstract ideas more tangible and engaging. Robotics fosters computational thinking, problem-solving skills, and algorithmic logic, all of which are fundamental to computer science (Egbert et al., 2021). Moreover, it aligns with 21st-century skills by promoting creativity, collaboration, and communication. Robotics can cater to diverse learning styles, allowing for personalized and differentiated instruction. Additionally, it prepares students for future careers by providing practical experience in programming, engineering, and automation (Misirli & Komis, 2023). By introducing robotics, educators can create a dynamic learning environment that supports not only computer science standards but also interdisciplinary connections, diversity and inclusivity, and the development of critical skills for the digital age.

Assessment

Assessments in the field of computer science, overseen by the STEAM teacher, have transitioned to digital formats. Using platforms such as CS First and anecdotal records for Computer Science for San Francisco, these digital assessments provide a streamlined and efficient means of evaluating student performance and progress. However, it is important to note that the district currently maintains no records or data related to computer science assessments, reflecting a gap in the tracking and documentation of student achievements in this domain. The school keeps extensive records of all other education data. This includes a digital black box of student assessment data. These assessment data include aMath, aReading, CBMR, and FAST.

Additionally, grade levels keep extensive data on all common grade-level assessments in math and reading.

Professional Development

The initiation and organization of professional development initiatives related to computer science have predominantly stemmed from the proactive efforts of the STEAM teacher, usually taking place during the summer and outside regular school hours. Although individualized professional development opportunities have been facilitated throughout the current school year, the district as a whole has not yet been allocated dedicated time for collective growth in this domain. Unfortunately, there are currently no scheduled plans for computer science professional development within the district for this academic year. The district has one full professional development day per month.

The Need to Integrate

In today's ever-changing world, the integration of computer science can further enrich educational opportunities for students and enhance their preparedness for the future. Students can gain skills like creativity, critical thinking, problem-solving, communication, and cooperativity skills within the scope of computational thinking (Sen et al., 2021). Computer science needs to be integrated into the Iowa Falls School District curriculum to align with the school's profile and its mission to empower all students to learn, lead, and succeed. Incorporating computer science education aligns with Iowa Falls School District's mission to prepare students for success both locally and globally. Computer science fosters essential 21st-century skills, including problem-solving, critical thinking, creativity, and adaptability, which are crucial for students to thrive in a rapidly evolving world.

An extensive plan provides consistency in computer science education throughout the district. An integrated framework means that all teachers across the district follow a standardized computer science curriculum. This consistency ensures that students at every grade level encounter the same fundamental concepts and skills. By having a well-defined curriculum in place, the district can guarantee that students are acquiring essential computer science knowledge. This whole-system solution establishes clear learning outcomes for computer science at every grade level. These outcomes are aligned with the district's goals and mission. With consistent learning objectives, students and their parents know what to expect from the computer science program, creating transparency and accountability. Learning objectives, sometimes referred to as learning outcomes, play a pivotal role in facilitating effective learning. They serve to articulate the expected student competencies resulting from instructional activities, thereby contributing to the development of more efficient instructional planning, learning activities, and assessment methods (Gronlund, 2000).

Consistency in computer science education ensures that all students, regardless of their grade level or the teacher they are assigned, have an equal opportunity to access and benefit from computer science instruction. It eliminates disparities in learning experiences that may occur when only one teacher is responsible for computer science education. This type of curriculum and instruction would enable the district to implement standardized assessment and evaluation methods. Teachers could assess student performance more objectively, leading to fair and reliable evaluations. These assessments can be used to measure progress and inform instructional improvements. Teachers can share their experiences, best practices, and insights, leading to a more informed and enriched computer science curriculum. Collaborative lesson planning and curriculum design can lead to more engaging and relevant learning experiences. Johnson (2014) found that when teachers work together to create engaging lessons, students are more likely to be motivated and involved in their learning. This collaborative environment helps maintain a high standard of education.

With a systematic solution, the district can provide a holistic computer science education that covers a broad range of topics and skills. This breadth of knowledge ensures that students receive a comprehensive understanding of the subject, preparing them for a wide range of future opportunities as well as ensuring that computer science education aligns with district and state standards. The solution would help the district meet its educational obligations and prepare students to meet the academic benchmarks set by relevant authorities.

These approaches to computer science integration ensure that all students in the Iowa Falls School District receive a standardized and high-quality education. This approach eliminates inconsistencies, promotes fairness, facilitates teacher collaboration, and aligns with district and state standards. By maintaining a consistent level of exposure and instruction, the district can effectively prepare students for success in the 21st century. Tran (2018) highlights the importance of leveraging computer science access in diverse elementary classrooms to promote young students' computational thinking; motivation in computer science topics; and the learning of essential soft skills such as collaboration, persistence, abstraction, and creativity to succeed in today's digital world.

School Data and Analysis

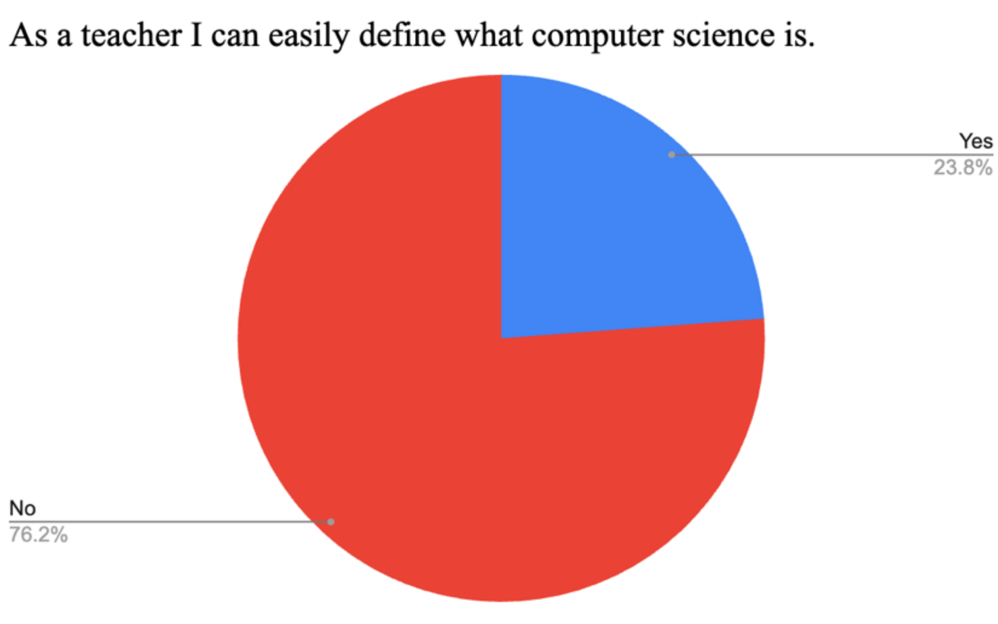
Data Collection

Survey data was collected from classroom teachers during the 2023-2024 school year at Iowa Falls Elementary Schools. These educators range from kindergarten to fifth grade. Teachers were surveyed on their knowledge of computer science and their feelings related to the concepts and implementation. This survey was conducted through a Google Form survey and was anonymous so teachers could provide honest responses. Murdoch et al (2014) found that survey participants were more likely to give honest feedback if surveys were truly anonymous.

Teachers were asked to assess their capability to define the concept of computer science. The data portrays the teachers' responses to this inquiry, categorized into two groups: "Yes" and "No." (See Figure 1.) Over 76% of the responding 17 teachers answered "No," indicating that they encountered difficulty in defining computer science. By contrast, only four teachers expressed confidence in their ability to define this term. The responses reveal that a substantial number of teachers may need additional training and professional development opportunities to improve their understanding of computer science.

Figure 1

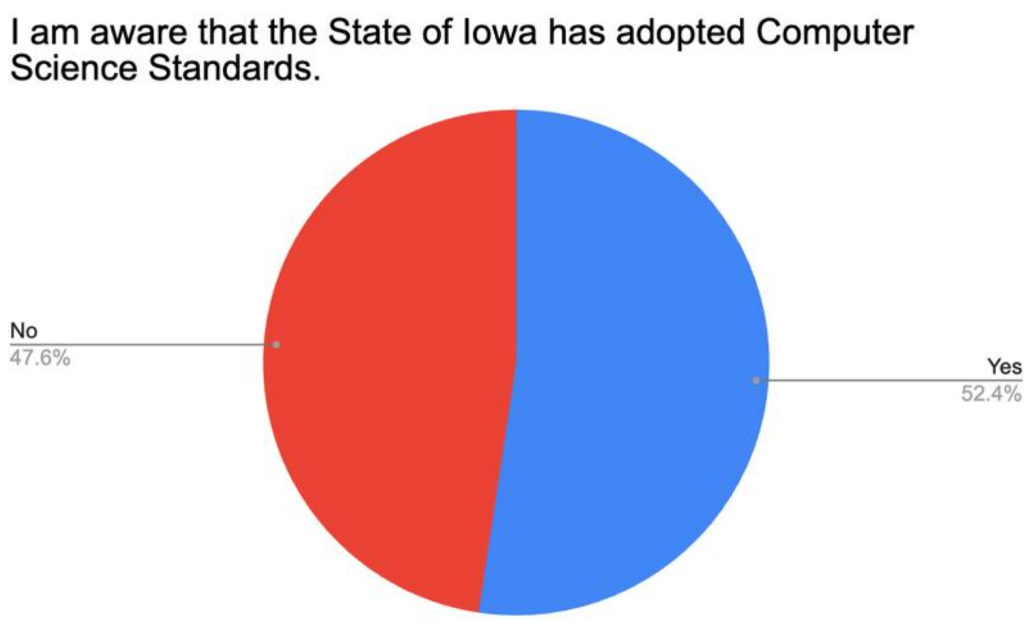
Current Teacher Knowledge



Ten of the 21 teachers (52.4%) are aware that the State of Iowa has adopted computer science standards. (See Figure 2.) By contrast, eleven individuals are not aware of the adoption of Computer Science Standards in the State of Iowa. This group may require more information or education regarding the state's expectations on computer science in the curriculum. The awareness of the adoption of Computer Science Standards is an important factor as it influences how educators, schools, and stakeholders approach computer science education in Iowa. Those who are aware of these standards are likely to be better positioned to incorporate computer science into their teaching and curriculum planning. For those who are unaware, efforts to disseminate information and provide training may be necessary to promote the effective implementation of computer science education in the state.

Figure 2

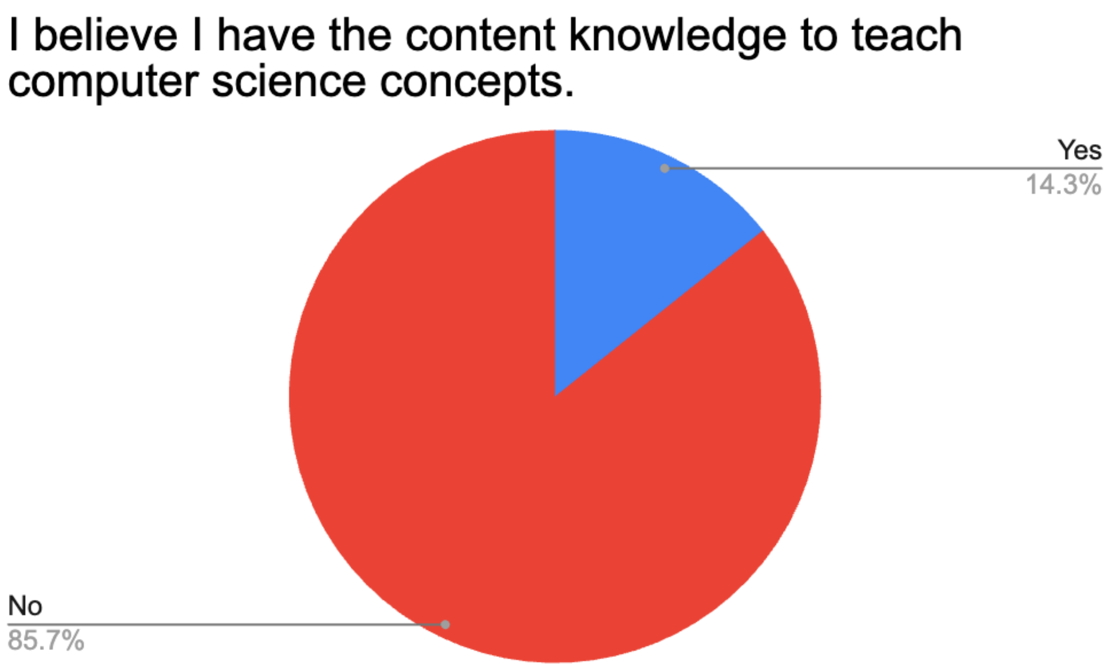
Awareness of Standards



Three teachers expressed confidence in their content knowledge to teach computer science. (See Figure 3.) They believe they have the requisite expertise to effectively instruct in this subject. A significant majority of 18 responded with "No," indicating that they do not believe they possess the content knowledge to teach computer science. This result suggests that most educators surveyed lack confidence in their ability to instruct in this field. This data underscores the varying levels of confidence among respondents in their content knowledge to teach computer science. While a small number are self-assured, the majority do not feel adequately prepared. This result highlights the importance of providing professional development, training, and resources to support educators in building their content knowledge and confidence in teaching computer science effectively.

Figure 3

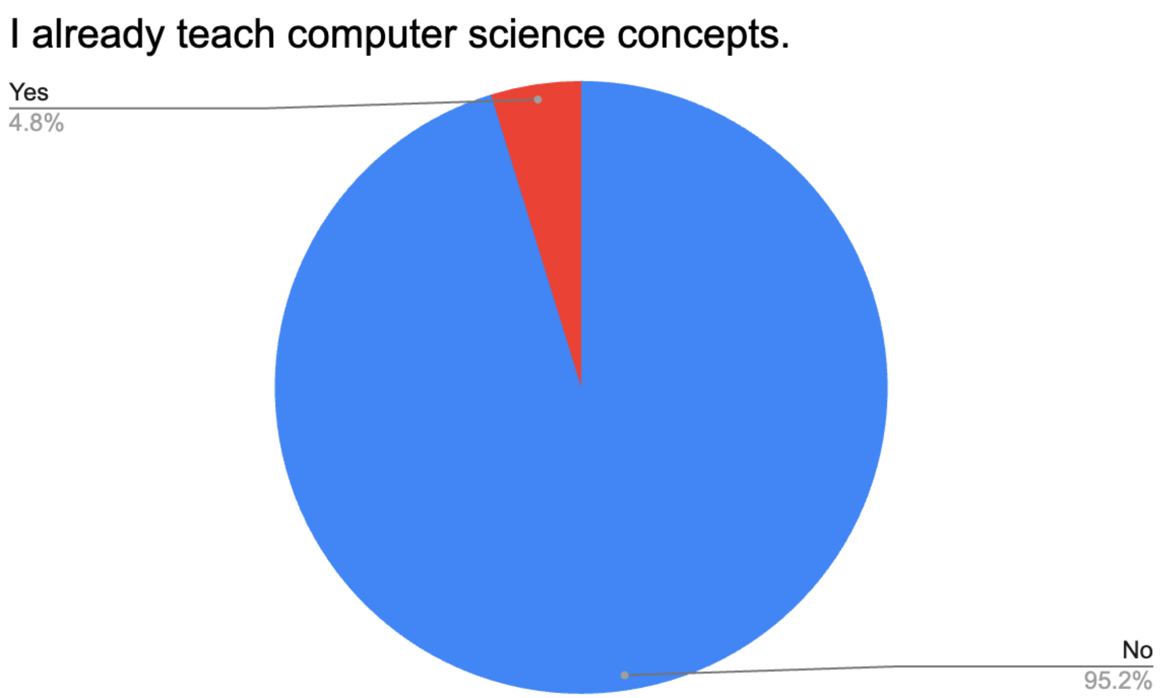
Current Teacher Competence



A high percentage (95.2%), specifically 20 individuals out of 21, indicated that they are not currently teaching computer science concepts. (See Figure 4.) Only one educator indicated that they are already implementing computer science in the classroom. This data underscores the prevalence of non-instruction in computer science concepts among the surveyed individuals, with only one individual actively engaging in teaching these concepts. These responses present an opportunity to promote and expand the teaching of computer science concepts, potentially through sharing knowledge, resources, and best practices to encourage further integration of computer science in educational settings.

Figure 4

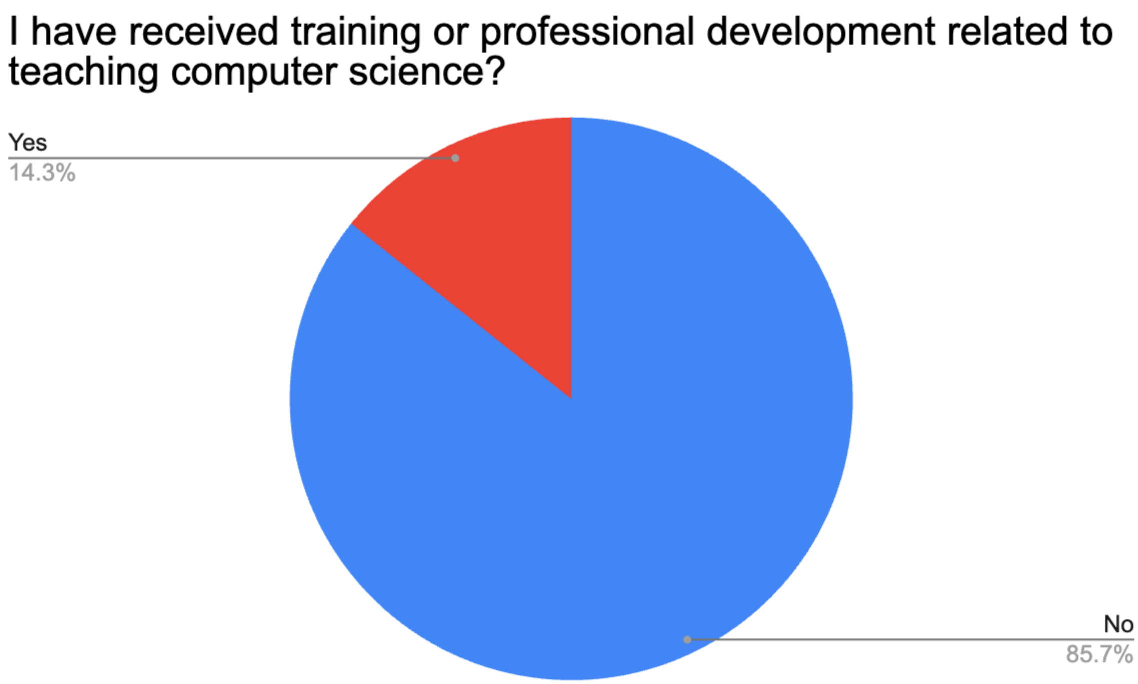
Current Classroom Practices



When asked if they have received training or professional development related to teaching computer science or coding, 85.7% responded with “no.” (See Figure 5.). These individuals lack formal preparation in this area. By contrast, three teachers responded with "Yes," signifying that they have indeed received training or professional development in teaching computer science or coding. This data underscores the importance of providing more opportunities for educators to receive professional development in computer science and coding as this can enhance their effectiveness in teaching these subjects. Sharing best practices and resources among educators can further promote the growth of computer science education in educational settings.

Figure 5

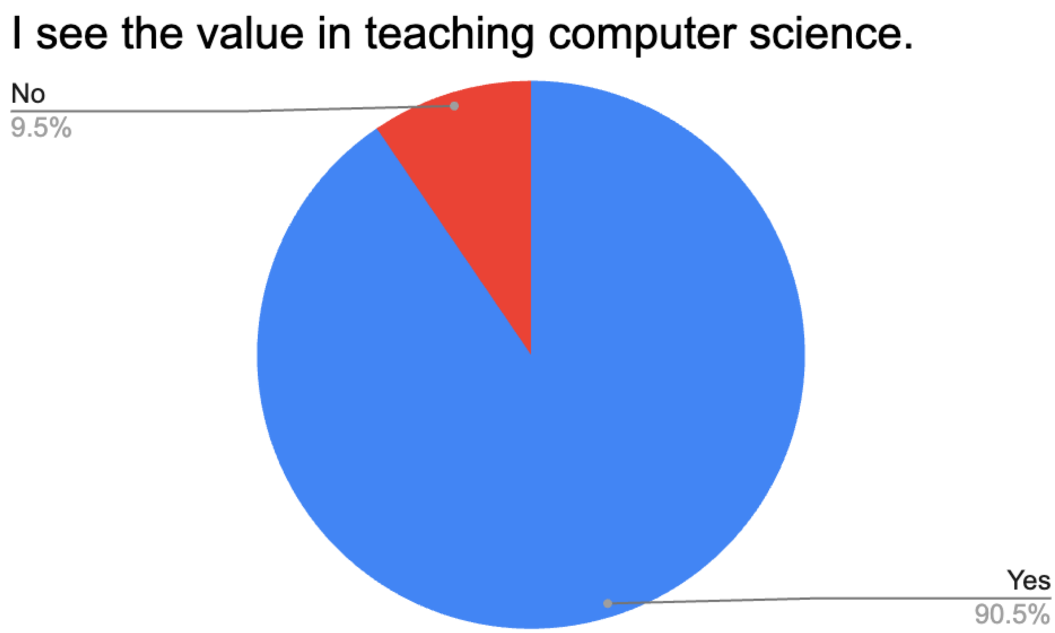
Professional Development Practices



The data in Figure 6 reveals that the majority of respondents, specifically 19 teachers out of 21, recognize and appreciate the value of teaching computer science. They understand the importance and relevance of incorporating computer science education into the curriculum, seeing it as a valuable component of modern learning. In contrast, only two responded with "No," suggesting that they may not fully recognize the value or relevance of teaching computer science in an educational context.

Figure 6

Recognizing the Significance of Teaching Computer Science



The survey data collected from Iowa Falls Elementary School teachers during the 2023-2024 school year provides valuable insights into the current state of computer science education in the elementary grades. It reveals that a significant number of teachers struggle to define computer science and lack the confidence to teach it effectively. Additionally, a majority of teachers are not aware of the computer science standards adopted by the State of Iowa, and most are not actively teaching computer science concepts in their classrooms. Furthermore, many teachers have not received training or professional development in computer science or coding.

However, the data also shows that the majority of respondents recognize the value of teaching computer science.

In terms of implications for a school improvement plan, it is evident that there is a pressing need for comprehensive professional development programs to enhance teachers' knowledge and confidence in computer science education. The plan should focus on providing training, resources, and best practices to support educators in building their content knowledge and teaching skills. It should also prioritize raising awareness of Iowa's computer science standards among teachers and school stakeholders. Moreover, the plan should support the integration of computer science into the curriculum by sharing knowledge and encouraging collaboration among teachers. By addressing these needs, the school can better equip its educators to effectively teach computer science and ensure that this valuable subject becomes an integral part of the educational experience for students.

Action Plan

The vision of this action plan is to equip students K-5 with comprehensive computer science skills and knowledge to empower them for success in the technologically evolving world. We will need to establish a systematic approach to integrate computer science into the curriculum across all grade levels, fostering a supportive learning environment and providing educators with the necessary tools, resources, and professional development. Below are the necessary steps to integrate computer science k-5 into the school curriculum.

Strategies

A diverse array of strategies was thoroughly considered during the formulation of this school improvement plan. The process involved a comprehensive examination and evaluation of multiple methodologies and approaches. Strategies were meticulously reviewed, ranging from pedagogical enhancements and curriculum enrichment to professional development initiatives and resource allocation. Furthermore, a detailed analysis encompassed diverse models employed in educational settings, including successful paradigms from other districts, contemporary research findings, and innovative practices in the field of education. This comprehensive approach allowed for a well-informed and multifaceted strategy to be constructed, one that aims to address various facets of education and foster an inclusive, dynamic improvement plan for the school.

This computer science integration plan leverages the experience and expertise available through the Iowa STEM Computer Science is Elementary website (Iowa STEM, 2019). The plans from other school districts on this platform provided a wealth of information, including successful strategies, best practices, and innovative approaches employed by fellow educators and administrators in similar educational settings. Resources there could be adapted to the unique context and requirements of Rock Run and Pineview Elementary. Moreover, the collaborative nature of the Iowa STEM Computer Science is Elementary website encourages networking and knowledge sharing with other education professionals, fostering a collective effort towards the enhancement of computer science education in our district. This invaluable resource played a pivotal role in shaping this plan as it ensured that the approach was not only well-informed but also aligned with contemporary practices and the evolving landscape of computer science education in Iowa (Iowa STEM, 2019).

Embedded curriculum and standalone curriculum represent two distinct approaches to integrating computer science into the educational framework. Embedded curriculum intertwines computer science concepts within existing subjects, seamlessly blending coding, computational thinking, and problem-solving skills into various disciplines like math, science, or even art. This method leverages the existing coursework to infuse computational principles organically, offering a contextualized learning experience. In contrast, a standalone curriculum refers to separate, dedicated courses solely focused on computer science, allowing in-depth exploration and dedicated study of coding, programming, and technological concepts. Standalone courses often offer comprehensive knowledge and skill development specific to computer science but may lack the interdisciplinary context that embedded curriculum provides. While standalone courses offer in-depth specialization, the embedded curriculum facilitates a more holistic understanding by weaving computer science organically into various subjects, allowing students to perceive the practical application of computational thinking in diverse contexts.

In pursuit of an effective computer science curriculum, this plan integrates Computer Science for San Francisco as a standalone program due to its comprehensive alignment with CSTA (Computer Science Teachers Association) standards. This curriculum not only meets educational benchmarks but also offers a ready-to-implement structure, providing complete lessons. Notably, one of the primary advantages is its cost-free availability, relieving the district of financial constraints. Furthermore, in tandem with this standalone curriculum, the integration of grade-level robotics into the embedded curriculum has been a strategic decision. Kindergarten students will engage with Bee Bots, 1st graders will explore Indi, and 2nd graders will delve into Dash robots. This progression continues with 3rd graders engaging with Ozobots, 4th graders with Finch Robots, and 5th graders with drones.

This multi-tiered approach aims to immerse students in practical, hands-on experiences aligned with their developmental stages, fostering a cohesive understanding of computer science concepts through a diverse range of robotics experiences. Hughes et al. (2022) revealed a substantial increase in students' learning achievements when using the project-based learning method integrated with robotic visual programs in comparison to the traditional teaching method integrated with paper practice teaching materials. The project-based learning approach, coupled with robotic visual programs, demonstrated significantly higher effectiveness in enhancing students' learning outcomes.

Year One

In the context of a school improvement plan, the collected survey data strongly indicates a critical need for comprehensive professional development initiatives aimed at enhancing teachers' proficiency in computer science education. (Again, see Figure 5.) To effectively address this gap, the establishment of the Computer Science Leadership Team (CSLT) is necessary in our school. This group should include personnel representing each grade level, an elementary STEAM instructor, an instructional coach, principals, and the technology integrationist. Its purpose is to direct the district's mission to integrate computer science into the curriculum. The leadership team will attend an intensive three-day computer science boot camp given that teacher training has a high impact on success. According to Acevedo-Borrega (2022) teaching of computer science concepts requires pure knowledge of the concepts as a whole, as well as an educator who is open to the process of learning and reflection. This training will help to increase competencies in the diverse spheres of computer science.

A two-hour professional development in August will serve as an introductory discourse for all teachers. The focus will be on the benefits, purpose, and expectations of the imminent computer science integration plan. The CSLT will meet monthly to sustain progress and uphold a dedicated focus on computer science education. These regular gatherings will involve discussions about strengths and areas that require improvement in our elementary program.

Notably, critical decisions concerning purchases and curriculum will be deliberated and decided upon at these meetings.

In September, the initiation of the implementation will bring a focused and detailed review of Unit 1 from the Computer Science for San Francisco curriculum (CS for SF) by the CSLT. This examination is strategically designed to prepare the groundwork for embedding computer science concepts into the elementary educational framework. The CSLT will analyze Unit 1 to ensure that it aligns with the existing curriculum, effectively introducing and developing a structured approach to teaching computer science across all grade levels. This review will involve planning, aligning learning objectives, and designing pedagogical methods tailored to each grade level's requirements. In October Unit 1 will be introduced and integrated into classroom settings across the elementary curriculum. This integration aims to ensure a smooth and comprehensive introduction of computer science concepts into the educational experience of students, fostering a strong foundation for their technological learning journey.

November will start with a STEM night event to foster community support by inviting students, parents, educators, and local community members to participate. It offers an opportunity for students to demonstrate and share their acquired knowledge and skills in STEM and computer science with their families and the community. Families and community members attending the event will gain a deeper insight into the educational experiences of students, allowing them to actively support their children's academic pursuits. Furthermore, such events provide a platform for educators to showcase the value and importance of STEM and computer science education in preparing students for the modern, technology-driven world.

Additionally, community support is encouraged through potential partnerships with local businesses, tech companies, and STEM professionals. These entities can actively participate in or sponsor the event, showcasing real-world applications of STEM and computer science, offering mentorship opportunities, and providing insights into potential career paths for students. By uniting students, families, educators, and community members in a collaborative and celebratory environment, a STEM night fosters a sense of collective support and enthusiasm for STEM and computer science education within the community. November will wrap with the implementation of Unit 2 of CS for SF by the STEAM Teacher. Given this educator’s prior experience implementing this program, classroom teachers will not need to be overwhelmed with new material to teach. The second unit involves hands-on work with computer science concepts that vary by grade level.

Recognizing the significance of empowering all teachers with the skills necessary to integrate technology into their respective subjects, the professional development session in December will feature a focused module on teaching robotics to educators across all disciplines, including specialized subjects. Led by the STEAM teacher and the technology integrationist, this session aims to equip every teacher, including specials, with hands-on experience involving various robotic resources like Bee-Bots, Indi Robots, Dash Robots, Ozobots, Finch Robots, and Drones. This session ensures that all teachers possess the proficiency to seamlessly incorporate these resources into their teaching methodologies, enabling the integration of technology across the educational spectrum.

In January teachers will convene for a two-hour professional development session where they will scrutinize the CSTA standards, aligning their pedagogical practices with the foundational tenets of computer science education. Teachers will also start the process of unpacking power standards. In February and March the CSLT will orchestrate the implementation of Unit 3 by the STEAM Teacher. This unit includes programming from scratch.

During this unit the team will visit elementary schools, observing firsthand the complete integration of the prescribed curriculum.

In April and May, the CSLT will craft embedded lesson examples, anticipating a comprehensive three-hour professional development session for all teacher levels. This collaborative endeavor will aim to weave two embedded lessons into the existing curriculum, laying the groundwork for a transformed educational landscape, with plans to implement in the following year. See Figure 7 below for a comprehensive plan of year one.

Figure 7

Year One Monthly Plan

Month	**Task**	**Person(s) Responsible**
June/July	• Develop a Computer Science Leadership Team (CSLT) that includes one teacher from every grade level, an elementary grade level tag, an **elementary level STEAM teacher**, an **instructional coach, elementary principals, and the technology integrationist**. Team leads are in bold • CSLT will attend a Computer Science Camp (3 days of intense computer science training) • Monthly CSLT team meeting	Principals should approach and select grade-level members of the CSLT CSTL team leads will coordinate camp

August	• 2-hr. professional development for all teachers at the beginning of the year discussing benefits, purpose, and expectations. • Monthly CSLT team meeting	Planned by Team Leads
September	• 2-hr professional development on Unit 1 of CS for SF • Monthly CSLT team meeting	STEAM teacher/ Instructional coach
October	• Implementation of Unit 1- CS for SF • Monthly CSLT team meeting	All classroom teachers
November	• National STEM Day CS activities in STEAM • Monthly CSLT team meeting • Unit 2 CS for SF	STEAM Teacher
December	• 2-hr professional development for teachers to explore resources: Bee-Bots, Indi Robots, Dash Robots, Ozobots, Finch Robots, and Drones.	Put on by STEAM Teacher and Tech integrationist
January	• 2-hr professional development to look at CSTA standards/ • Monthly CSLT team meeting • Unit 3 CS for SF	All elementary teachers and CS standards. STEAM Teacher

February	• CS leadership team, visit elementary school with full integration • Monthly CSLT team meeting	CSLT
March	• Unit 3 CS for SF • Monthly CSLT team meeting	STEAM Teacher
April	• Prepare mock embedded lesson examples for classroom teachers • Monthly CSLT team meeting	CSLT
May	• Model embedded lesson examples for grade-level teachers • 3 hr professional development for all grade-level teachers to develop 2 embedded lessons to fit in current curriculum. • Development of Power Standards	CSLT Team leads CSLT

Year Two

The second year of our computer science implementation plan will continue to foster a comprehensive environment for technological and computational learning. All teachers will participate in a two-hour professional development session, reinstating the core benefits, purpose, and anticipated outcomes of our computer science program. During this session, we will also share embedded plans and allocate dedicated team planning time, ensuring a unified and cohesive approach to integration. In year two teachers will implement the two embedded activities planned the previous year. This will happen at grade-level teams' discretion.

September will bring another two-hour professional development session to delve into Unit 2 of Computer Science for San Francisco. Our monthly meetings with the CSLT remain a cornerstone, providing a forum for insights, strategizing, and planning. November sees us celebrating National STEM Day with engaging computer science activities embedded within our STEAM education framework. Simultaneously, our 2nd Annual STEM Night will offer an enriching experience for students, parents, and community members, showcasing the diverse applications and joys of STEM and computer science.

December will celebrate Computer Science Week, highlighting the achievements and progress made in this area. Meanwhile, the New Year brings a renewed focus on assessment strategies, with grade-level teams designing rubrics for power standard assessments. This commitment further solidifies our dedication to aligning with Computer Science standards. At this point in year 2 the STEAM teacher will give control and implementation of CS for SF Unit 2 to classroom teachers.

February again brings the visit of the CSLT offering support and encouragement to the elementary school with its ongoing full integration of computer science education. March witnesses the complete implementation of Unit 3 of Computer Science for San Francisco from Kindergarten to 5th grade by the STEAM teacher. The instructional rounds in April will serve as a unique opportunity for modeling lessons or skills that teachers feel need more attention, fostering an environment of mutual support and learning. Finally, in May, a comprehensive four-hour professional development session for all grade levels will concentrate on developing three embedded lessons that integrate into our current curriculum.

As the second year unfolds, these strategic actions aim to further fortify our commitment to comprehensive computer science integration, fostering an environment of continuous learning, collaboration, and innovation among our staff and students. The concerted efforts of the CSLT, alongside the dedicated participation of teachers and the support of the entire educational community, will ensure a robust and impactful implementation of computer science education at our institution. With one year remaining in this plan, we are near complete integration. (See Figure 8 below.)

Figure 8

Year Two Monthly Plan

Month	**Task**	**Person(s) Responsible**
August	• 2-hr. professional development for all teachers at the beginning of the year RE-discussing benefits, purpose, and expectations. Also, share embedded plans and team planning time • Monthly CSLT team meeting • Implementation of 2 embedded activities will occur at Grade-level team discretion • Teacher Satisfaction Survey	Planned by Team Leads
September	• 2-hr professional development on Unit 2 of CS for SF	Put on by STEAM teacher

	• Monthly CSLT team meeting	
October	• Implementation of Unit 1 • Monthly CSLT team meeting	
November	• National STEM Day CS activities in STEAM • 2nd Annual STEM Night • Monthly CSLT team meeting	STEAM Teacher
December	• Implementation of Unit 2 CS for SF • Celebrate Computer Science Week	Classroom Teachers Designed by CSLT celebrated by all teachers
January	• 2 hr. pd-Grade-level teams will create a rubric for power standard assessment • Monthly CSLT team meeting	All elementary teachers and CS standards.
February	• CS leadership team, visit elementary school with full integration. • Monthly CSLT team meeting	
March	• Unit 3 CS for SF K-5 Implementation	STEAM Teacher
April	• Instructional Rounds to model lessons or skills that teachers feel like they need more work on	CSLT

	• Monthly CSLT team meeting	
May	• 4 hr. professional development for all grade levels to develop 3 embedded lessons to fit in current curriculum. • Monthly CSLT team meeting	CSLT Team leads All Teachers

Year Three

The third year of the computer science integration plan brings a full integration of computer science education. This year proceeds with continued professional development sessions, particularly focusing on Unit 3 of the Computer Science for San Francisco curriculum. Teachers will engage in a study of the concepts outlined in Unit 3, delving into the curriculum's core components to develop a comprehensive understanding. Unit 1 implementation, guided by grade level teachers will take place again in October. Celebrating the National STEM Day activities in November and hosting the 3rd Annual STEM Night again provides a platform to showcase students' acquired knowledge in STEM and computer science, enhancing community engagement and promoting the value of these educational domains.

The subsequent months spotlights the sequential implementation of Units 2 and 3, as well as a celebration of Computer Science Week. In the spirit of progression, the year concludes with a 2-hour professional development session discussing the way forward on a continued path of integration. Throughout this period, the CSLT team meets monthly, steering the progress and development of the comprehensive computer science integration plan. The academic year concludes by conducting surveys to gather essential feedback on teacher satisfaction.

For an overview of Year 3, see Figure 9.

Figure 9

Year Three Integration Plan

Month	**Task**	**Person(s) Responsible**
August	• 2-hr. professional development for all teachers at the beginning of the year RE-discussing benefits, purpose, and expectations. Also, share embedded plans and team planning time. • Monthly CSLT team meeting • Implementation of 5 embedded activities will occur at Grade-level team discretion (2 from year 1, 3 from year 2) • Teacher Satisfaction Survey	Planned by Team Leads
September	• 2-hr professional development on Unit 3 of CS for SF • Monthly CSLT team meeting	Put on my STEAM teacher
October	• Implementation of Unit 1 • Monthly CSLT team meeting	Grade Level Teachers

November	• National STEM Day CS activities in STEAM • 3rd Annual STEM Night • Monthly CSLT team meeting	STEAM Teacher
December	• Implementation of Unit 2 CS for SF • Celebrate Computer Science Week	Classroom Teachers Designed by CSLT celebrated by all teachers
January	• 2 hr. pd- Next Steps, how we can continue on a path of integration • Monthly CSLT team meeting	All elementary teachers and CS standards.
February	• CS leadership team, visit elementary school with full integration. • Monthly CSLT team meeting	CSLT
March	• Unit 3 CS for SF K-5 Implementation	Classroom Teachers
April	• Instructional Rounds • Monthly CSLT team meeting	CSLT
May	• End-of-Year Survey Teachers • Monthly CSLT team meeting	CSLT Team leads All Teachers

This action plan presents a structured and detailed strategy for the integration of computer science education into the K-5 curriculum a Rock Run and Pineview Elementary. It uses a multi-tiered approach, combining a standalone curriculum and embedded activities, to offer a comprehensive learning experience. Leveraging insights from successful educational practices and resources, the plan ensures the alignment of the curriculum with CSTA standards while promoting hands-on learning experiences at various grade levels. This three-year plan systematically unfolds the progression from foundational teacher training to the full integration of curriculum and community engagement events. The careful design ensures a continuous learning trajectory for educators, facilitating their growth and proficiency in delivering computer science education.

The phased approach across three years signifies the dedication to a sustained and evolving integration, fostering an environment where both students and educators actively participate in the dynamic landscape of computer science education. With a focus on collaborative efforts and consistent improvement, this action plan serves as a blueprint to cultivate a holistic understanding of computer science concepts, equipping students for success in our technology-driven world. The commitment to regular assessments, monthly CSLT meetings, and ongoing professional development underlies the dedication to refining and enhancing the integration of computer science education over the years.

Implementation of the Plan

Timeline

For the successful integration of this computer science plan, a detailed timeline was created. (See Table 7-9 above.) School leaders need to approve this plan before the end of the 2023-2024 school year. Members of the CSLT will be chosen no later than April 2024 to ensure proper time and planning for the extensive pre-service three-day computer science training to take place in June or July. During the school year ongoing monthly meetings will be conducted to ensure that implementation of the plan is successful. District leaders will continue to be involved throughout the plan to ensure consistency.

Identifying Resources

The implementation of a computer science education program for K-5 students requires a diverse array of resources. It necessitates comprehensive curriculum materials like Computer Science for San Francisco and online resources that align with CSTA standards. Additionally, access to technology resources—computers, tablets, and a variety of robotics such as Bee-Bots, Indi Robots, Dash Robots, Ozobots, Finch Robots, and Drones—is essential for teaching coding and programming concepts effectively. Although the district has a small number of each, it may be necessary to expand the number of robotics pieces. Professional development is crucial, involving training sessions, workshops, and continuous educational support for teachers to enhance their understanding and teaching skills in computer science. Needed training could easily be accomplished during the district's monthly professional development days. Community support is also valuable, requiring engagement with parents, local businesses, tech companies, and STEM professionals to offer mentorship programs, real-world connections, and insights into potential career paths for students. Assessment tools such as rubrics and surveys should be developed and implemented to evaluate the program's effectiveness and measure teachers' satisfaction. Adequate budget allocation is essential to cover the costs of technology resources, professional development, curriculum materials, and other necessities, ensuring the successful implementation of the program.

Staff Responsibilities

Administrators and District Leaders

In this comprehensive plan for integrating computer science education into the K-5 curriculum, administrators play a critical and multifaceted role. Their involvement is pivotal in ensuring the success and effective implementation. Principals provide the overarching leadership required for the successful execution of the plan. They oversee and guide the implementation process, ensuring that the plan aligns with the overall vision and goals of the district and school. They are responsible for resource allocation, ensuring that the necessary tools, materials, and technological resources are available for teachers and students to effectively integrate computer science education. These responsibilities include budgeting for any required purchases and supporting the acquisition of necessary tools or software.

Administrators play a crucial role in organizing and facilitating professional development opportunities. They ensure that teachers receive the necessary training, workshops, and resources to build their skills in teaching computer science concepts effectively. They are also responsible for monitoring the progress of the plan's implementation and conducting periodic evaluations to assess its effectiveness. They identify areas of success and any challenges that need addressing, making necessary adjustments to ensure the plan's success. District leaders also include the school board. Board members play a crucial role in supporting the initiative through policy advocacy, budget approval, and alignment with educational goals. Their endorsement ensures that the program receives the necessary financial and organizational backing.

Central Office

The central office oversees the district-wide implementation, providing logistical support, coordinating professional development, and ensuring curriculum alignment. They work closely with the Computer Science Leadership Team (CSLT) to address challenges, refine strategies, and sustain the long-term vision of computer science integration. Collaboration among students, the school board, and the central office creates a synergistic approach, fostering an environment where computer science education becomes an integral part of the educational landscape.

The central office also plays a key role in facilitating collaboration between educators, coordinating resources, and monitoring the overall progress of the computer science integration plan, ensuring its alignment with district-wide educational objectives.

STEAM Teacher

The STEAM teacher is instrumental in integrating computer science concepts into the existing curriculum across various grade levels. They develop and structure the computer science curriculum in alignment with the standards and educational benchmarks, ensuring that it complements and enhances the overall educational framework. The STEAM teacher is also responsible for teaching computer science concepts to students, using resources such as the Computer Science for San Francisco curriculum and other materials. The STEAM teacher implements the planned lessons, engaging students in various coding activities, computational thinking, and problem-solving exercises. As a core member of the Computer Science Leadership Team (CSLT), the STEAM teacher takes an active leadership role. They participate in planning, decision-making, and overseeing the effective implementation of the computer science curriculum across grade levels. They are involved in collecting data related to student progress and success in learning computer science concepts. Additionally, they assist in assessing the effectiveness of the implemented curriculum and making necessary adjustments based on assessment outcomes. They will also model best practices in teaching computer science concepts through demonstrations, classroom examples, and strategies that effectively engage and teach students.

Students

Students are central to the success of the computer science integration plan, actively engaging in a transformative learning experience. Their role involves participating in hands-on activities, immersing themselves in the embedded and standalone computer science curriculum, and showcasing their knowledge during events like STEM Night. As the primary beneficiaries, students provide valuable feedback through assessments, contributing to the continuous improvement of the program. Their involvement extends beyond the classroom, fostering a sense of enthusiasm for STEM and computer science within the school community. By participating in National STEM Day activities and the annual STEM Night, students become ambassadors of the program, demonstrating the real-world applications of computer science concepts to their families and the wider community. The plan places students at the forefront, aiming to empower them with the skills and knowledge needed for success in the technologically evolving world.

Teacher Feedback

Assessing teacher experience will give invaluable feedback regarding the integration of computer science education into the K-5 curriculum. The survey will be conducted at the end of every implementation year using Google Forms to help ensure ease and access and participation. The survey will include a variety of topics, including feelings on professional development, student outcomes, and overall feedback. The responses obtained from teachers will serve as a vital data collection point, providing insights into the effectiveness of the program, teacher readiness, the program's impact on student learning, areas of improvement, and resource requirements. The survey is an essential tool to understand the challenges and successes of integrating computer science education, informing tailored adjustments, professional development, and resource allocation to meet the specific needs of teachers.(see Appendix B)

The survey's data will be used by the CSLT to aid in making informed decisions about the optimization and enhancement of the computer science curriculum, fostering an environment for continuous improvement and growth in this educational domain.

Barriers and Challenges

The integration plan presents several foreseeable challenges. Among these, teacher preparedness stands out, as not all educators may possess the necessary expertise or comfort in teaching technology and computer science, potentially leading to resistance to integrating these new concepts into their teaching methodologies Time constraints also emerge as a significant hurdle, as teachers already manage a packed curriculum, struggling to balance existing subjects with the integration of computer science education. Next, resistance to change, common with any new educational program, could further complicate the integration process among teachers, students, and parents. Continual professional development requirements for educators and the development of effective assessment strategies aligned with the computer science curriculum are additional hurdles that need to be addressed. Lastly, students' diverse backgrounds and varying levels of prior exposure to technology may create disparities in learning.

Conclusion

The State of Iowa has embraced the importance of computer science education, implementing standards in 2018 and enacting a law in 2020 to ensure all schools offer quality computer science education by July 2022. Lihui Sun (2022) supports the positive impact of computer science education in K-12 settings, promoting computational, creative, and critical thinking skills. In today's digital era, effective teaching of computer science is essential in education, focusing on fostering computational thinking skills and digital literacy. Various studies reviewed methodologies and emerging trends to ascertain the most effective approaches for promoting computer science as a core educational discipline.

Mouza et al. (2022) research indicates the significance of structured professional development programs for teachers to gain knowledge of computer science content, pedagogy, and technology. The findings emphasize the importance of providing teachers with ongoing and well-designed professional development opportunities to build their capacity for teaching CS effectively. Tran (2018) found that in-school computer science coding lessons positively impacted elementary-aged students, improving their understanding of concepts and promoting equity by reaching underrepresented students in economically disadvantaged districts. The study highlighted the importance of teacher knowledge, pedagogical practices, and embedding soft skills in the curriculum. These points were taken into consideration while crafting the computer science integration plan for Rock Run and Pineview Elementary.

With guidance from Iowa STEM, this three-year action plan for the integration of computer science education in K-5 curricula outlines a strategic approach focused on comprehensive teacher training, curriculum alignment, and community engagement. Combining a standalone curriculum and embedded activities, the plan aims to immerse educators and students in a multifaceted learning experience. By leveraging insights from contemporary educational practices and resources, the design ensures alignment with CSTA standards and hands-on learning experiences across different grade levels. While the plan demonstrates a clear progression from foundational training to curriculum integration and community events, it acknowledges challenges such as teacher readiness, resource constraints, and the need for continual professional development. Emphasizing collaboration, consistent assessment, and regular teacher feedback, the action plan outlines a framework fostering the future of computer science education.

The effectiveness of this comprehensive action plan for integrating computer science education in K-5 depends on a vigilant approach toward the various barriers and challenges identified. By acknowledging and proactively addressing issues such as teacher readiness and the need for ongoing professional development, this plan can be implemented successfully. Monitoring these challenges with a dedicated focus will ensure that the strategies remain adaptive and aligned with the evolving landscape of education, thereby fostering the successful integration of computer science into the school curriculum.

References

Acevedo-Borrega, J., Valverde-Berrocoso, J., & Garrido-Arroyo, M. D. C. (2022). Computational thinking and educational technology: A scoping review of the literature. *Education Sciences*, *12*(1), 39. https://doi.org/10.3390/educsci12010039

Arend, M. (2022, May). Governor Kim Reynolds on Iowa's next chapter. *Site Selection*. Retrieved September 1, 2023 from https://siteselection.com/issues/2022/may/governor-kim-reynolds-on-iowa-s-next-chapter.cfm

Arfé, B., Vardanega, T., Montuori, C., & Lavanga, M. (2019). Coding in primary grades boosts children's executive functions. *Frontiers in Psychology*, *10*. https://doi.org/10.3389/fpsyg.2019.02713

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130-145. https://doi.org/10.1016/j.compedu.2019.04.013

Computer Science. (n.d.). Iowa Department of Education. Retrieved September 20, 2023, from https://educateiowa.gov/pk-12/instruction/computer-science

Computer Science Skills: Computational Thinking Explained. (2021, January 5). Retrieved September 30, 2023, from https://csforca.org/computer-science-skills-computational-thinking-explained/

Computer Science Teachers Association (2017). *CSTA K–12 Computer Science Standards, Revised 2017*. Retrieved September 30, 2023 from https://csteachers.org/k12standards/

CS First. (2022). CS First by Google. Retrieved October 13, 2023, from https://csfirst.withgoogle.com/c/cs-first/en/welcome-to-cs-first/overview.html

CS for SF. (n.d.). CS for All in San Francisco. Retrieved October 13, 2023, from https://www.csinsf.org/

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in the early years of primary education. *Computers & Education*, *150*, 103832. https://doi.org/10.1016/j.compedu.2020.103832

Department of Education, Education Department [281] notice of intended action, H.R. 2020-ARC5207C (Iowa Oct., 2020).

Egbert, J., Shahrokni, S. A., Abobaker, R., & Borysenko, N. (2021). "It's a chance to make mistakes": Processes and outcomes of coding in 2nd grade classrooms. *Computers & Education*, *168*, 104173. https://doi.org/10.1016/j.compedu.2021.104173

Explore Census Data. (2021). United States Census Bureau. Retrieved October 13, 2023, from https://data.census.gov/

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem-solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, *63*, 87-97. https://doi.org/10.1016/j.compedu.2012.11.016

Gronlund, N. E. (2000). *How to write and use instructional objectives* (6th ed.). Merrill.

Hughes, C. E., Dieker, L. A., Glavey, E. M., Hines, R. A., Wilkins, I., Ingraham, K., Bukaty, C. A., Ali, K., Shah, S., Murphy, J., & Taylor, M. S. (2022). RAISE: Robotics & AI to improve STEM and social skills for elementary school students. *Frontiers in Virtual Reality*, *3*. https://doi.org/10.3389/frvir.2022.968312

Iowa Falls Vision/Mission/Commitments. (2022). Iowa Falls Schools. Retrieved October 12, 2023, from https://www.ifacadets.net/iowa-falls-27

Iowa Department of Education. (2022, June 20). *Computer Science State Plan*. Retrieved September 20, 2023, from https://educateiowa.gov/documents/computer-science-state-plan#:~:text=Computer%20Science%20State%20Plan.pdf

Iowa Department of Education. (2022). Iowa School Performance. Retrieved October 13, 2023, from https://www.iaschoolperformance.gov/ECP/StateDistrictSchool/SchoolDetails?DetailType=Proficiency&k=13499&y=2022

Iowa STEM. (2019, August). Retrieved November 3, 2023, from https://iowastem.org/cselementary

Johnson, D. W., & Johnson, R. T. (1999). Making cooperative learning work. *Theory Into Practice*, *38*(2), 67-73. https://doi.org/10.1080/00405849909543834

Lee, A. (2015). Determining the effects of computer science education at the secondary level on STEM major choices in postsecondary institutions in the United States. *Computers & Education*, *88*, 241-255. https://doi.org/10.1016/j.compedu.2015.04.019

Liu, C.-C., Cheng, Y.-B., & Huang, C.-W. (2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education*, *57*(3), 1907-1918. https://doi.org/10.1016/j.compedu.2011.04.002

Misirli, A., & Komis, V. (2023). Computational thinking in early childhood education: The impact of programming a tangible robot on developing debugging knowledge. *Early Childhood Research Quarterly*, *65*, 139-158. https://doi.org/10.1016/j.ecresq.2023.05.014

Mouza, C., Codding, D., & Pollock, L. (2022). Investigating the impact of research-based professional development on teacher learning and classroom practice: Findings from computer science education. *Computers & Education*, *186*, 104530. https://doi.org/10.1016/j.compedu.2022.104530

Murdoch, M., Simon, A. B., Polusny, M. A., Bangerter, A. K., Grill, J. P., Noorbaloochi, S., & Partin, M. R. (2014). Impact of different privacy conditions and incentives on survey response rate, participant representativeness, and disclosure of sensitive information: A randomized controlled trial. *BMC Medical Research Methodology*, *14*(1). https://doi.org/10.1186/1471-2288-14-90

National Center for Educational Statistics. (2023). National Center for Educational Statistics. Retrieved September 20, 2023, from https://nces.ed.gov/ccd/schoolsearch/school_detail.asp?Search=1&InstName=Rock+Run+Elementary&City=Iowa+Falls&SchoolType=1&SchoolType=2&SchoolType=3&SchoolType=4&SpecificSchlTypes=all&IncGrade=-1&LoGrade=-1&HiGrade=-1&ID=191473000907

Prasetiyo, W. H., Naidu, N. B. M., Sari, B. I., Mustofa, R. H., Rahmawati, N., Wijaya, G. P. A., & Hidayat, O. T. (2021). Survey data of internet skills, internet attitudes, computer self-efficacy, and digital citizenship among students in Indonesia. *Data in Brief*, *39*, 107569. https://doi.org/10.1016/j.dib.2021.107569

Sen, C., Ay, Z. S., & Kiray, S. A. (2021). Computational thinking skills of gifted and talented students in integrated STEM activities based on the engineering design process: The case of robotics and 3D robot modeling. *Thinking Skills and Creativity*, *42*, 100931. https://doi.org/10.1016/j.tsc.2021.100931

Sun, L., Guo, Z., & Zhou, D. (2022). Developing k-12 students' programming ability: A systematic literature review. *Education and Information Technologies*, *27*(5), 7059-7097. https://doi.org/10.1007/s10639-022-10891-2

Tran, Y. (2018). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research*, *57*(1), 3-31. https://doi.org/10.1177/0735633117743918

Tsortanidou, X., Daradoumis, T., & Barberá, E. (2019). Connecting moments of creativity, computational thinking, collaboration and new media literacy skills. *Information and Learning Sciences*, *120*(11/12), 704-722. https://doi.org/10.1108/ils-05-2019-0042

Vajen, B., Kenner, S., & Reichert, F. (2023). Digital Citizenship Education: Teachers' perspectives and practices in Germany and Hong Kong. *Teaching and Teacher Education*, *122*, 103972. https://doi.org/10.1016/j.tate.2022.103972

Yang, W., Ng, D. T. K., & Su, J. (2023). The impact of story-inspired programming on preschool children's computational thinking: A multi-group experiment. *Thinking Skills and Creativity*, *47*, 101218. https://doi.org/10.1016/j.tsc.2022.101218

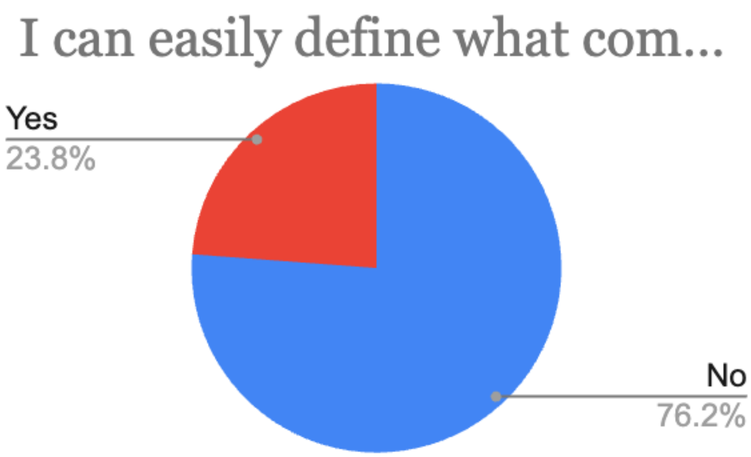
Zeng, Y., Yang, W., & Bautista, A. (2023). Computational thinking in early childhood education: Reviewing the literature and redeveloping the three-dimensional framework. *Educational Research Review*, *39*, 100520. https://doi.org/10.1016/j.edurev.2023.100520

Zhong, J., & Zheng, Y. (2023). "What it means to be a digital citizen": Using concept mapping and an educational game to explore children's conceptualization of digital citizenship. *Heliyon*, *9*(9), e19291. https://doi.org/10.1016/j.heliyon.2023.e19291

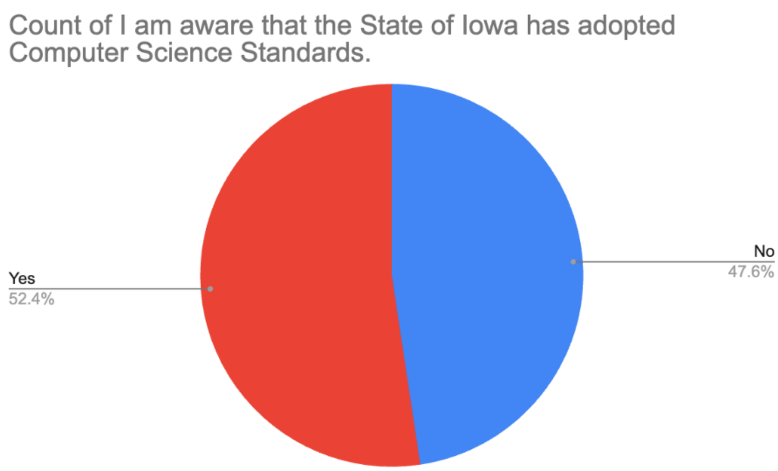
Appendix A

Pre-Integration Survey Results

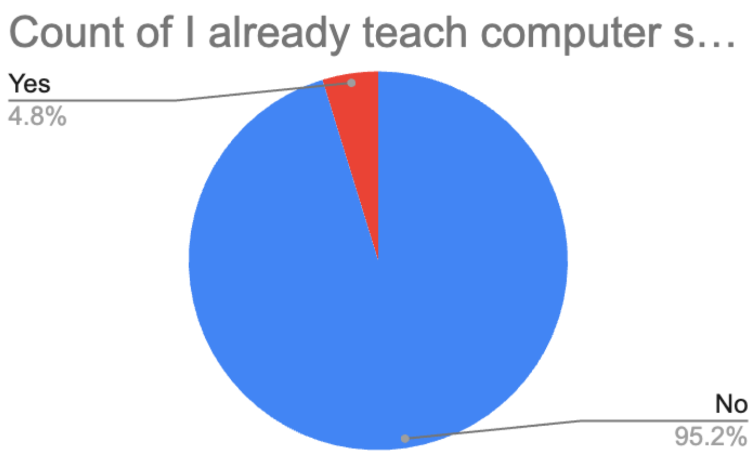
1. I can easily define what computer science is.



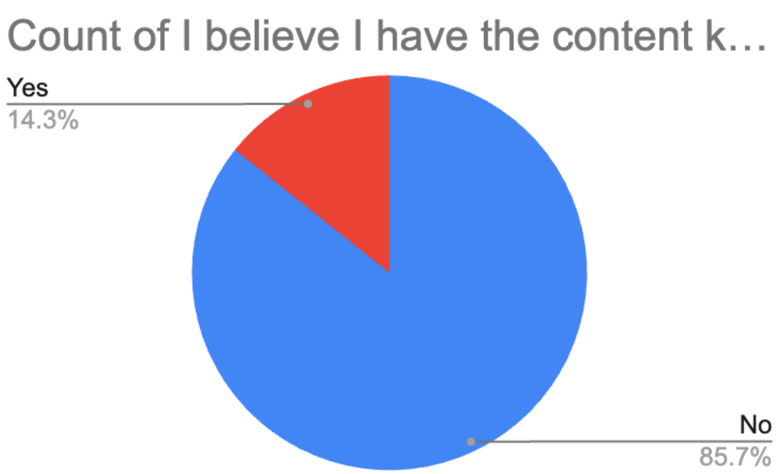
2. I am aware that the State of Iowa has adopted Computer Science Standards.



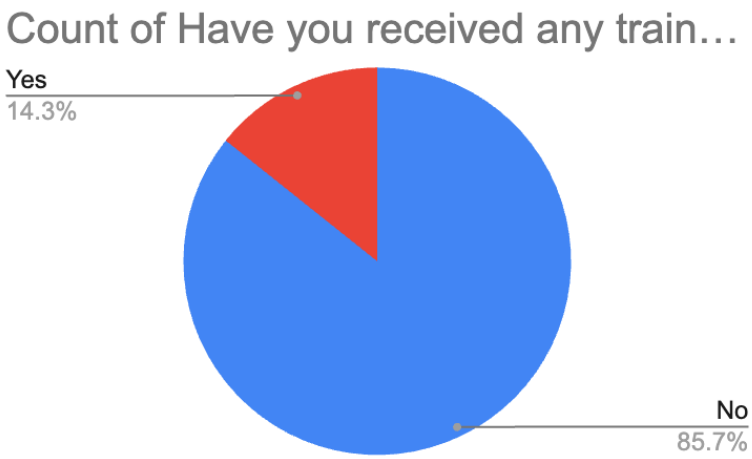
3. I already teach computer science concepts.



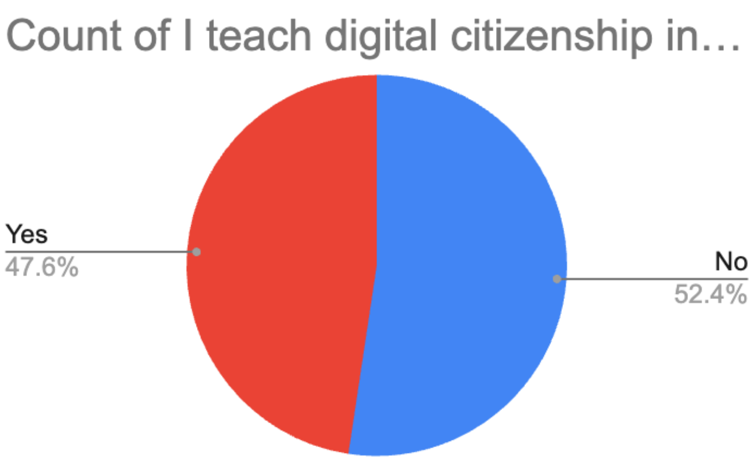
4. I believe I have the content knowledge to teach computer science.



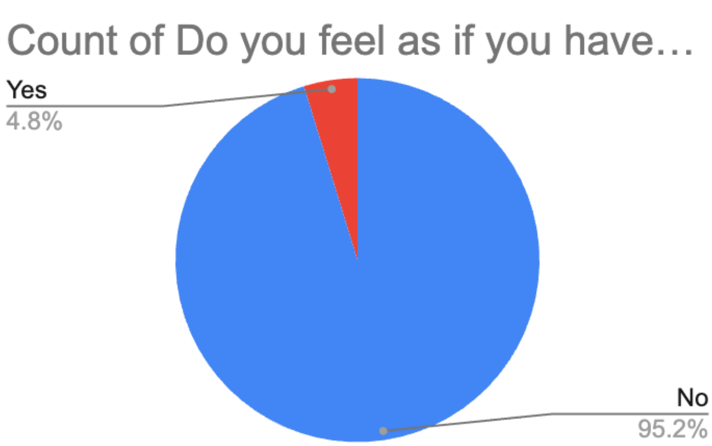
5. Have you received any training or professional development related to teaching computer science or coding?



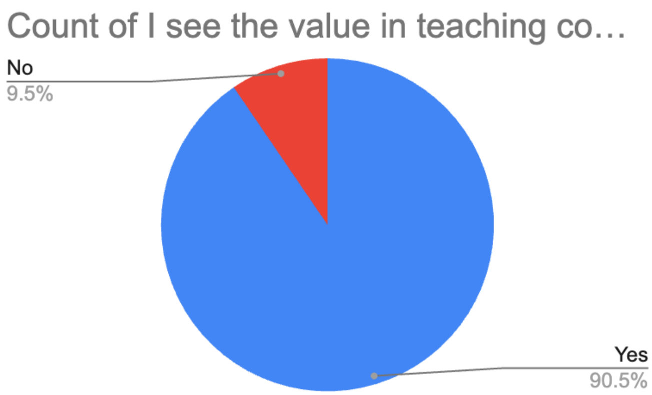
6. I teach digital citizenship in my classroom.



7. Do you feel as if you have the time and resources required to integrate computer science into your curriculum?



8. I see the value in teaching computer science.



Appendix B

End-of-Year Teacher Survey

Teacher Survey for Integration

“Your feedback is invaluable to ensure effective implementation and continuous improvement of the computer science integration program. Please take a few minutes to complete this survey. Thank you for your participation.”

Demographic Information:

1. Name: [Optional]
2. Grade Level: [Kindergarten / 1st Grade / 2nd Grade / 3rd Grade / 4th Grade / 5th Grade]
3. Years of Teaching Experience:
4. Have you received any prior training in computer science education? [Yes / No]
5. How comfortable do you feel in integrating technology into your teaching practices? [Scale: 1-5, 1 being uncomfortable, 5 being very comfortable]

Program Evaluation:

6. How well do you think the current computer science integration program aligns with the needs of the students and curriculum requirements? [Scale: 1-5, 1 being poor alignment, 5 being excellent alignment]
7. Which aspects of the computer science curriculum do you find most challenging to integrate into your classroom activities? (Select all that apply)
    - Coding
    - Computational Thinking
    - Robotics

- Problem Solving
- Others [Please specify]

8. Have you observed improvements in student engagement or understanding in subjects related to computer science since the program's implementation? [Yes / No]
    - If yes, please describe.
9. How would you rate the resources and support provided for the integration of computer science into your teaching? [Scale: 1-5, 1 being inadequate, 5 being excellent]
10. What additional resources or support would be beneficial to improve the integration of computer science in your classroom?

Professional Development and Training:

11. How effective have the professional development sessions been in preparing you to integrate computer science into your teaching? [Scale: 1-5, 1 being ineffective, 5 being highly effective]
12. What specific topics or areas related to computer science would you like to receive more training or support in?
13. Do you feel that the training received adequately covers the content and skills needed to teach computer science effectively? [Yes / No]

Student Outcomes:

14. Can you notice any observable changes in student understanding, creativity, or critical thinking skills after integrating computer science into your lessons? [Yes / No]

- If yes, please provide examples.

Overall Feedback:

15. What are the major benefits you've observed in integrating computer science into the curriculum?
16. What challenges have you faced in integrating computer science into your teaching practices?
17. Any additional comments or suggestions for further improvement?

Thank you for participating in this survey. Your input is highly valued and will help in enhancing our computer science integration program.