Jl. of Technology and Teacher Education (2024) 32(4), 583–614

Integrating Computational Thinking Within and Across Disciplines in the Context of Teacher Professional Development

AMANDA NOLTE
University of Delaware, USA
anolte@udel.edu

HILARY MEAD
University of Delaware, USA
hmead@udel.edu

CHRYSTALLA MOUZA
University of Illinois Urbana-Champaign, USA
cmouza@illinois.edu

ROSALIE ROLÓN-DOW
University of Delaware, USA
rosa@udel.edu

SOTHEARA VENG
University of Delaware, USA
sotheara@udel.edu

LORI POLLOCK
University of Delaware, USA
pollock@udel.edu

Teachers' lack of knowledge about computational thinking (CT) and limited opportunities to incorporate CT in existing curricula pose unique challenges at the elementary level. Despite the crucial role of professional development (PD) in preparing elementary school teachers to integrate CT in classroom instruction, there is little research documenting PD programs that focus on integration in literacy and mathematics compared to other subject areas. In this work, we present a PD program that integrates CT with disciplinary content to support teachers as they integrate CT with literacy and mathematics in elementary school classrooms. Using data from multiple sources, we present findings from two case studies to examine the impact of PD on teachers' integration of CT with content in lesson planning and implementation. Findings have implications related to the integration of CT in elementary school and teacher PD.

Keywords: computer science, elementary school, teacher professional development

INTRODUCTION

Recognition of the need for computer science (CS) education in K–12 schools has driven various reform initiatives and the development of CS curriculum standards for many years now, but much of the existing research focuses on the role of CS education at the secondary level (e.g., Qian et al., 2018; Sadik et al., 2020). Recently, however, there has been increasing attention given to the important role of CS education in the elementary grades, frequently by focusing on a set of underlying cognitive skills known as computational thinking (CT; Bers et al., 2022). Elementary schools are a natural starting point for CT because the early years of schooling play a crucial role in developing positive attitudes toward computing among young students (Ching et al., 2018).

Despite the importance of integrating CT in school curricula, teacher education programs have not traditionally provided opportunities for teachers to gain CT content knowledge (DeLyser et al., 2018; Mouza et al., 2021). As a result, incorporating those skills in existing curricula poses unique challenges, especially at the elementary level (Rich & Hu, 2019; Rich et al., 2021). In order to prepare elementary teachers to integrate CT in core disciplinary content, professional development (PD) that directly addresses these challenges is necessary (Vivian & Falkner, 2019). For instance, while elementary school schedules dedicate a majority of time to high-accountability subjects such as literacy and mathematics, teachers who are equipped to integrate CT in core content instruction can overcome limited opportunities to incorporate CT in required curricula (Century et al., 2020; Shehzad et al., 2023; Waterman et al., 2020).

Despite the crucial role of PD in preparing elementary school teachers to integrate CT, there is little research documenting PD programs that focus on integration in literacy and mathematics compared to other subject areas, such as science (e.g., Coenraad et al., 2022; Waterman et al., 2020). This work presents a PD program that integrates CT with disciplinary content to support teachers as they integrate CT with literacy and mathematics in elementary school classrooms. Specifically, the structure of the PD delivered during one academic year is described and the following research question is explored through two qualitative case studies: *What depth of CT integration is evident in teachers' content area lesson planning and instructional implementation?*

LITERATURE REVIEW

Computational Thinking Integration into Core Subjects

CT is often characterized by a set of problem-solving skills and practices (e.g., pattern recognition, decomposition, abstraction, algorithms) commonly utilized in CS that could be applied to other situations (Israel et al., 2015; ISTE & CSTA, 2011; Wing, 2006). Teaching CT skills can help students enhance their understanding of disciplinary content and become adept problem-solvers in various aspects of their lives, both professionally and as citizens (Gretter & Yadav, 2016). As such, benefits are not exclusive to those who go on to pursue careers in computing (Wing, 2006). Yet marginalized students often lack equitable access to CT opportunities (Fletcher et al., 2021), which can hinder their sense of belonging in computing (Cheryan et al., 2015; Master et al., 2021). In fact, research suggests that as early as third grade, students begin to endorse stereotypes of who belongs in computing (Master et al., 2021). Thus, identifying ways to introduce CT in elementary grades is essential for fostering early engagement and helping students envision themselves within the field. This approach can set the foundation for CS education in secondary schools and help broaden participation in the field.

A prime obstacle to effectively engaging elementary students in CT is time (Feng & Yang, 2022). Research repeatedly demonstrates that time constraints hinder elementary teachers' ability to incorporate CT in their required curricula (Repenning et al., 2015; Rich & Hu, 2019; Rich et al., 2021). Elementary teachers typically focus on literacy and mathematics due to increased accountability in these subjects, leaving less time for other areas (Century et al., 2020). Therefore, an effective strategy for incorporating CT into elementary schools is to adopt an integrative approach (Waterman et al., 2020), which allows teachers to leverage their current content and teaching expertise (Israel et al., 2015; Rich & Hu, 2019). Moreover, an integrative approach demonstrates the relevance of CT in other disciplines and in students' own lives, and can thus promote accessibility and transfer (Ryoo, 2019; Shehzad et al., 2023). Finally, an integrative approach is equitable in that it allows students to access CT even in schools that may not have the resources to offer independent computing classes (Yadav et al., 2016).

Much of the existing research related to the depth and effectiveness of CT integration focuses on STEM subjects due to the increased dependence on computational tools and methods in contemporary science (Aslan et al., 2024) as well as the role of CT in standards associated with science and mathematics, including the Next Generation Science Standards (NGSS Lead States, 2013) and the Common Core State Standards (CCSS; National Governors Association Center for Best Practices & Council of Chief State School officers, 2010). Weintrop et al. (2015), for instance, posited a reciprocal relationship between mathematics and CT and developed a taxonomy of practices that employ mathematics as a meaningful context to situate concepts and practices of CT. Other studies have shown that integrating CT concepts can enhance students' interest and motivation in learning mathematics and promote collaboration among students (Ke, 2014; Lambic, 2011; La Paglia et al., 2017). Moreover, in contrast to traditional teaching methods, where teachers primarily deliver information, teachers who integrated CT have assumed alternative roles during CT activities, often serving as facilitators and guides (Ardito et al., 2014; Hershkowitz et al., 2023).

Although prior work on integrating CT into literacy is limited, studies have demonstrated that the integration of CT and literacy activities can be beneficial for students (e.g., Bers et al. 2022; Burke & Kafai, 2010, 2012; Pektas & Sullivan, 2021; Whyte et al., 2019). Whyte et al. (2019) examined the design and implementation of a six-week intervention in the context of an after-school program and its role in supporting students' development of both computing and literacy skills. Findings indicated that such integration supported novice learners in developing both CT skills (e.g., algorithm design) and literacy skills (e.g., visualizing narrative structure). A number of students, however, did not produce complete stories, suggesting the need to carefully consider learning activities and instructions. These findings point to the need for more work that examines the integration of CT in elementary literacy. Our study contributes toward addressing this gap by investigating CT integration in elementary literacy, as well as mathematics, through teacher PD.

Computational Thinking Tools for Mathematics and Literacy

Other than time, a key barrier to integrating CT within elementary grades is teacher preparation, as efforts to introduce pre-service teachers to CT through either a standalone course (e.g., Mouza et al., 2017) or integration in content and methods courses have only recently emerged (Margulieux et al., 2022; McGinnis et al., 2020). Yet, as Jocius et al. (2023) demonstrated, teachers can and do grow in their ability to integrate CT into disciplinary teaching through effective PD. Our PD program, described below, likewise incorporated high quality PD features as well as a variety of CT activities and tools, both technological and "unplugged." A central focus of the PD was Scratch, a block-based programming language, which allows users to create interactive stories, animations, and games. Scratch was chosen because it is accessible, free, and can be integrated into different core subjects. Introducing teachers to 'low floor' activities, like block-based programming, can alleviate the initial unease or fear associated with CT (Adler & Kim, 2018).

Although Scratch is not explicitly intended to teach mathematics, it does incorporate essential mathematical ideas within its programming environment (Taylor et al., 2010). Scratch can be used to reinforce concepts such as geometry, algebra, and problem-solving (e.g., Sjöberg et al., 2018; Taylor et al., 2010). Additionally, Scratch works well for literacy due to its narrative approach and tools that support the creation of stories, such as visual graphics, characters, settings, and programming blocks. Indeed, CT concepts inherent in Scratch can be aligned with the CCSS for writing (e.g., Smith & Burrow, 2016). For example, the concept of "sequence," referring to identifying a series of steps for a task, is related to the standard for using temporal words to signal event order.

Research suggests that the use of Scratch can positively contribute to students' learning of mathematics and literacy. Calao et al. (2015) found that, compared to sixth grade students who received traditional mathematics instruction, students who used Scratch as part of a mathematics class improved their performance of key mathematical processes associated with modeling, reasoning, problem-solving, and exercising (i.e., comparing and executing procedures and algorithms). The authors argue that exercising is a skill that is less developed in traditional math classes, but especially benefits by the use of programming. In elementary school specifically, Iskrenovic-

Momcilovic (2020) found that third grade students using Scratch for learning basic geometric shapes outperformed students who studied using more traditional practices.

In literacy, studies indicate that the use of Scratch can effectively enhance writing skills. Fields et al. (2014) studied how students collaboratively created interactive stories using Scratch, received feedback, and showcased their work. They found that this process led to more complex story designs and improved coding skills. Similarly, Pektas and Sullivan (2021) demonstrated that Scratch supported fourth-grade students in mastering their state standard related to narrative writing within a formal school setting.

Albeit useful, most studies to date focus on students' learning experiences without adequate attention to teachers' perspectives. This study aims to provide insights into the integration of Scratch in elementary school contexts from teachers' perspectives by examining how they approached its integration in their content area to support students' CT development.

CONCEPTUAL FRAMEWORK

Although a number of frameworks have emerged to guide the integration of CT in content area instruction, they primarily focus on defining and synthesizing CT skills for K–12 students (e.g., Mills et al., 2021) or providing diagnostic tools about educators' readiness for CT integration (Education Development Center, 2024). Our study is informed by a framework for examining the depth of integration of CT in elementary content area lessons that was proposed by Waterman et al. (2020) and revised by Coenraad et al. (2022). We chose this framework because of its direct focus on examining *depth* or *levels* of CT integration, which provided a valuable data analytic lens. Waterman et al. identify three levels of integration that informed their efforts to integrate CT in existing elementary science curriculum:

1. Exist: At the *exist* level, CT is already present in the lesson and "can simply be called out or elaborated upon" (Waterman et al., 2020, p. 54).
2. Enhance: At this level, integration is characterized by the "creation of additional tasks or lessons to *enhance* the disciplinary concept and provide clear connection to computing concepts that are present, but not central, to the existing lesson" (Waterman et al., 2020, p. 55).

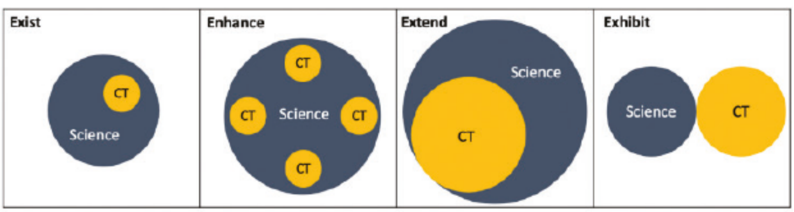
3. Extend: At the deepest level of integration, "new lessons or sequences of lessons *extend* the disciplinary concept as a basis for CS exploration" (Waterman et al., 2020, p. 55).

The three categories are positioned in a hierarchy, with each subsequent level representing a greater depth of integration.

Coenraad et al. (2022) later applied the framework to explore the depth of CT integration demonstrated in science lesson plans created by teachers following participation in PD, but they found that some plans did not fit into any of the established categories. To capture the depth of integration present in these lesson plans, they added a fourth category to the framework, called *exhibit*. At the *exhibit* level, lessons "used a CT activity, typically programming, to exhibit science knowledge gained through other means" such as "a book or online research" (Coenraad et al., 2022, p. 13). Although Waterman et al. (2020) explicitly arrange their categories according to the increasing depth of integration, Coenraad et al. do not position the *exhibit* level within the hierarchy. Instead, *exhibit* is simply presented as an additional category in which "CT and science are both present but do not overlap" (Coenraad et al., 2022, p. 13). They visually summarize the four levels of integration as shown in Figure 1.

Figure 1

Levels of CT Integration Presented in Coenraad et al. (2022, p. 13)



The framework aligns closely with the purpose of our study, which explores the depth of CT integration in teacher-created lessons following their participation in PD. While the studies conducted by both Waterman et al. (2020) and Coenraad et al. (2022) are situated in the context of elementary science, the overall intention of the framework is to support the development of CT within disciplinary contexts without compromising student learning in the underlying discipline. Thus, our work can contribute towards understanding the role of the framework in evaluating CT integration in disciplines outside of science. Further, we *extend the framework* beyond teacher-created lesson plans to explore its application to teachers' lesson implementations, thereby enhancing our understanding of CT integration in content area instruction at the elementary level.

CONTEXT AND METHODS

Description of PD Program

Existing research indicates that high quality PD is characterized by a focus on content, pedagogy, active learning, coherence, and sustained duration (e.g., Desimone, 2009; Rich et al., 2021). Our PD program was designed to reflect these characteristics. Specifically, the PD included a simultaneous focus on both CT and disciplinary content. As shown in Table 1, teachers were introduced to CT skills (Mills et al., 2021) and built their understanding of block-based programming and physical computing. At the same time, the PD included time to explore core content standards (e.g., mathematics and English Language Arts [ELA]) and how to integrate CT into these disciplines. Additionally, the PD program included exposure to and modeling of effective CT pedagogy (e.g., unplugged activities), as well as culturally responsive pedagogy (CRP), an asset-based approach that connects instruction to students' backgrounds, interests, and needs (Gay, 2018; Ladson-Billings, 1995). Further, teachers spent time engaging in hands-on activities with computational tools, such as Scratch (active learning), and were provided with opportunities to connect new ideas and practices from the PD to their existing curricula (coherence).

In terms of duration, the PD spanned six months: November to May. It included four two-hour Saturday workshops in the fall and winter and monthly individual meetings in the spring. During these meetings, teachers discussed the design of their instructional plans and received personalized support for their lesson implementations. This allowed teachers time to build their knowledge and apply new learning to practice.

Participants

Recruitment for the PD targeted elementary school teachers. The opportunity was disseminated to state, district, and school leaders and through professional networks (e.g., Computer Science Teachers Association). Five teachers attended at least one PD session. This number fell short of the participation target; due to the demands on elementary teachers during the pandemic, it was challenging to attract them to a weekend commitment. Three teachers completed all aspects of the PD: attended face-to-face sessions, designed and implemented lessons, and facilitated data collection in their classrooms. All three agreed to participate in the research and became our focal participants.

Table 1

PD Design

PD Sessions (November – February)			
	CS Content	Disciplinary Content	CRP
Session 1	Introduction to computing tools and unplugged activities	Aligning sample CT lessons with mathematics and literacy standards	Introduction to CRP: Role of personal identities in planning & instruction
Session 2	Programming in Scratch: Remixing	CT-integrated instructional resources	Modeling CRP in a CT-integrated lesson: Scratch Animate a Name
Session 3	Programming in Scratch: Classroom logistics	CT lesson from the perspective of a learner	Adapting CT-integrated lessons to incorporate CRP
Session 4	CT-integrated lesson planning: Exemplars, collaborative lesson planning, and whole group share-out		CRP strategies: Review & quick reference resource
Individual Meetings (February – May)			
Teacher Consultations/Support for Lesson Planning & Classroom Implementation			

Courtney[1] was a fifth-grade teacher who taught ELA and mathematics in a traditional K–5 public school in a diverse suburban setting. In her school, roughly three-quarters of the student population were students of color, and over a third were classified as low-income. On a pre-workshop survey, Courtney reported no prior experience with coding or technology-related PD.

Ana and Emma both taught at an independent, secular day school located just outside a major city, serving primarily white students (64%) in grades pre-K–12. Ana was a fifth-grade mathematics teacher, while Emma taught a standalone CS course which all fifth-grade students took as a ‘spe-

1 All participant names are pseudonyms.

cial.' This meant that all of Ana's students were also enrolled in Emma's CS class, which was the teachers' motivation for working together. Emma had the most coding background, with "considerable" experience in block-based programming and "some" experience in Java and Python. Ana was familiar with older programming languages from her previous career in technology.

A multiple case study design was employed, in which Courtney is the first case. Since Ana and Emma developed their plans in collaboration with each other and taught the plans sequentially to the same group of students, they together constitute the second case. Even though Ana and Emma did not co-teach, they treated CT and content integration as a joint goal. Case study was appropriate for our goals in order to illustrate particular instances of integrating CT in elementary content instruction and how teachers experienced integration (Merriam, 1998; Yin, 2017).

Data Collection

All data collection procedures were approved by the university's Institutional Review Board. Qualitative data were collected from four sources during the school year: (a) teacher lesson plans, (b) field notes and artifacts from lesson observations, (c) teacher interviews, and (d) student interviews. Most data were collected by team members who had also designed and facilitated the PD. This overlap in roles led to deep familiarity with the PD and the participants. To address potential bias, two additional team members were included in data collection and analysis. Specifically, a researcher not involved with the PD conducted all lesson observations, and one of the project's external evaluators participated in biweekly team meetings and contributed to data analysis.

Lesson plans: Teachers were given a lesson plan template, which asked them to identify lesson goals related to content area and CT learning, pedagogical practices, and lesson activities and assessments.

Observations: Unstructured observations (Mulhall, 2003) of teachers' lesson implementations were conducted by a research team member who was not involved with planning or facilitating the PD. During each observation, the researcher recorded field notes, which captured: (a) evidence of CT integration, as well as details relevant to sequencing and timing of lesson activities; (b) student engagement throughout the lesson; (c) teacher and student dialogue, including direct quotes where possible; and (d) the overall classroom environment, including the physical space, available technology, and class routines and expectations.

As noted, Courtney constituted one case, while Ana and Emma together constituted the second case. In the PD, teachers were asked to design and then implement a CT-integrated lesson with considerable flexibility in the length of the lesson.[2] Thus there are three differences in scope between the cases: (a) Courtney alone constituted one case, while Ana and Emma implemented lessons in their respective classrooms but together constituted a case; (b) Courtney's lesson took place on a single day, while Ana's and Emma's extended over multiple days; and (c) Courtney was a generalist teacher working with a single group of students (n=15), while Ana and Emma taught specific subjects (mathematics and CS) to three groups of students (n=53). Taken together, these differences meant there were more observations for Ana and Emma's lessons (n=7) compared to the single observation of Courtney. Nevertheless, the observations in each case were proportional to the amount of instruction that occurred and, therefore, to the opportunities for the research team to observe.

Teacher interviews: Semi-structured interviews (Merriam, 2009) were conducted with each of the three teachers and included 12 questions that focused on: (a) perceptions of the PD program, (b) experiences of planning and implementing CT-integrated content area lessons, (c) student responses to the lessons, and (d) needs for ongoing support. Each interview lasted approximately 30–40 minutes. Interviews were conducted via Zoom, recorded, and transcribed for analysis.

Student interviews: Semi-structured interviews were also conducted with a sample of students participating in each lesson. Courtney implemented her lesson with all students in her class (n=15). Of those, four had parent/guardian consent to participate in interviews, and all were included in the study. As noted, Ana and Emma implemented their lessons with three groups of students (n=53), many of whom returned consent forms. A sample of ten students was chosen for participation based on availability (i.e., free in the same class period). In her interview, Ana noted that she selected a range of students, including some who exhibited increased interest in computing and some who did not complete their work due to a lack of engagement.

Students were asked nine questions that focused on: (a) overall responses to the lesson, (b) opinions about using CT in a content area, (c) atti-

[2] This flexibility was provided by design, based on research that elementary teachers are significantly constrained by time and other curricular demands (Century et al., 2020; Feng & Yang, 2022). The team rationalized that if elementary teachers were to integrate CT at all, they needed flexibility in how they would do so.

tudes toward coding, and (d) connections between the lesson and their own lives. Each interview lasted approximately 10–15 minutes. Interviews were conducted via Zoom, recorded, and transcribed for analysis.

Data Analysis

Lesson plans and observations were analyzed using a deductive content analysis approach (Elo & Kyngäs, 2008). A deductive approach was appropriate given the research questions, which specified core content and CT as *a priori* categories of interest. One author analyzed lesson plans and field notes, treating each as a discrete data source. All references to core content and CT were annotated, as well as instances in which references to CT and core content co-occurred (see Table 2). The author then conducted a comparative analysis of corresponding lessons and observations to evaluate alignment of CT integration across planning and instruction. For the second case study, an additional layer of analysis was conducted to compare Ana and Emma's planning and implementations within the context of their lesson collaboration. The author wrote analytic memos throughout the comparative analysis and identified notable themes, which were then discussed with the research team.

Table 2

Sample Annotations of Field Notes Recorded During Emma's Lesson Implementation

Field Notes Excerpt	Annotation
Emma: "When you are in doubt, look at this [visual of geometric shape in the coordinate plane, shown on whiteboard]. You see how I draw a dot here? There is a little dot on the triangle, like in the math book, which shows which side of the shape is showing [when it is flipped]." [Emma] shows a poster and asks if the shape shown is a two-dimensional or three-dimensional object, and she demonstrates rotation and flipping the object.	Content (reviewing concepts related to geometric transformations)

Field Notes Excerpt	**Annotation**
[Emma] asks a student how they can demonstrate in Scratch how we can show the shape is flipped. The student says they can use a text box. The students will be moving shapes around the coordinate plane, and when it moves quadrants, there will be a portion of the animation that explains it. The goal is that students who know nothing about the topic create a sort of lesson within Scratch to watch this and understand quadrants, rotation, angles, vertices, and perhaps more.	Content + CT integration (creating animation and explanation of geometric transformations in Scratch)
"Remember there is only one shape, but it's going to have 4 costumes," Emma says as she walks around the room. She then goes to a student with his hand up and shows him how to change backgrounds. Emma: "Remember to have your backgrounds named and use 'switch backdrop to.'" Emma: "Guys, it's the same shape, but it's traveling right. You need to change costumes in each quadrant." Students are asking each other for clarification as to the number of backdrops they need.	CT (changing backgrounds and costumes in Scratch)[3]

Teacher interviews were coded and analyzed using Dedoose computer software (SocioCultural Research Consultants, 2021). Structured codes based upon prior work (Mouza et al., 2022) were used for the parent codes, or major ideas (see Table 3). Within each parent code, subthemes were identified using open coding, and constant comparisons were made within and across transcripts (Saldaña, 2021). Analytic memoing was used to distill assertions from the coded data. One author analyzed the teacher interviews and wrote memos, which were then shared and discussed with the full team.

Student interview data were organized by case and by question (i.e., all responses to each question from the two schools were compiled). Then, open coding was used to identify emergent themes. Instances of each theme were counted to identify their relative weight within each case. A sample excerpt from the codebook is shown in Table 4.

[3] While there are references to geometric shapes (content) in the excerpt, the content references are secondary to this step, which focuses on backgrounds and costumes.

Table 3

Parent Codes

Parent Code	Definition
Context	Information about the participant's job or school
PD positives	Things that participants liked best about the PD
PD deltas	Changes that participants recommended to the PD or challenges/difficulties they had with it
PD implementation	Ideas or resources that participants used in their classrooms
Student needs	Participant describes the needs of his/her students which may or may not be addressed by the PD
Support	Participant describes follow-up support received or desired from PD team in the past, present, or future

Table 4

Student Interview Coding Sample – Ana & Emma (10 interviews)

Interview Question	Codes (Frequency)
Can you tell me about the project you have been working on?	Mathematics content (9) CT content or tool (Scratch) (8) Other (1)
What did you like about it?	Coding/Scratch (4) Content area (mathematics) (4) Fun (4) Creative (3) Social (2) Easy (1)
What did you not like about it?	Nothing (4) Time it took (4) Complicated/hard (2)

FINDINGS

In this section, we describe the depth of integration evident in each case, following the Coenraad et al. (2022) framework on levels of CT integration. Within each case, we first present the depth of CT integration evident in the teachers' instructional *planning,* as demonstrated by the lesson plans themselves and teachers' accounts of their planning in interviews. We subsequently present the depth of CT integration evident in teachers' *implementations*, as demonstrated by observations, teacher interviews, and student interviews.

Case Study 1: Courtney

Planning

While Courtney taught both fifth grade ELA and mathematics, she chose to integrate CT in ELA. Courtney's lesson addressed content standards that required students to use narrative techniques to demonstrate characters' responses to situations. Drawing on a book the class previously read together, Courtney planned for students to engage with the content objectives by using Scratch to develop and animate dialogue that might occur between themselves and the main character. Her lesson plan achieved the PD goal that teachers integrate CT in content instruction.

In her interview, Courtney described CT as far "outside [her] comfort zone" and identified herself as a total beginner: "I really didn't know anything about computer science [or] programming." She chose Scratch as a tool to integrate because it was accessible to beginners: "I felt like the day [during the PD] we just focused on Scratch and did stuff, I was getting it pretty quickly, and I could see my kids getting it." Courtney also capitalized on existing instructional resources for Scratch, available through *CS First*[4], which a PD facilitator shared with her during an individual consultation. Courtney framed that resource as an effective support: "I wrote the lesson plan, but the computer science part was done for me. That's what I need. … You need something that's fast and easy to use." The duration of the PD, which included individual assistance following the initial PD sessions, facilitated Courtney's exposure to additional resources and aided her ability to coherently integrate CT in her required curriculum.

Courtney described herself as a teacher who preferred teaching ELA to mathematics, but her choice of content area for integration was more practical. She explained the relative difficulty of predicting her class pacing in mathematics, while ELA offered more flexibility in terms of skills and se-

---

[4] CS First provides CS curricula that make coding easy to teach: https://csfirst.withgoogle.com/s/en/home

quence: "I was like, I can't plan this lesson on volume and then … when I go to teach it, we're in fractions. I felt like pacing-wise, I can always teach an ELA lesson, [I] can always do a read-aloud." Courtney's lesson included a read-aloud of a mentor text, followed by an instructional video about how to use Scratch. The next day, students integrated these two strands as they created and coded dialogue between characters in Scratch.

Viewed independently, Courtney's lesson plan could be seen as representative of either *enhance* or *exhibit* according to the Coenraad et al. (2022) framework. In this case, the level of integration depends on whether students continued their ELA learning through the coding activity (*enhance*) or mastered the ELA objective before turning to Scratch (*exhibit*). However, Courtney's implementation, discussed below, clarifies that her lesson reflected the *enhance* level of integration because the CT-related activity helped her students develop a deeper understanding of narrative techniques used to write dialogue.

Implementation

Courtney's lesson implementation demonstrated the *enhance* level of CT integration (Coenraad et al., 2022). In her interview, Courtney discussed how CT tasks contributed to, rather than simply reinforced, students' learning of the content objective. For example, she recounted a conversation that illustrates how the CT activity enhanced students' understanding of narrative writing:

> My boys were like, 'I want to write, 'sup bro?'' and I'm like, 'Okay, that's fine. Write how you want it, like how would you really talk to your friend? You don't go to your friend and say, 'hello, do you want to play with me?' … Maybe you can't write 'sup' in your essay, but you can write 'sup' in dialogue.'

Here, students' reflections on how they animated themselves as Scratch sprites prompted deeper understandings of the ways in which narrative writing techniques differ from techniques employed when writing essays. In interviews, students agreed that the lesson contributed to their ELA content knowledge, sharing several ways they felt the project strengthened their writing. Specifically, students referenced skills related to planning (e.g., mapping dialogue), selecting vocabulary (e.g., to make dialogue interesting), and using appropriate punctuation (e.g., how to use quotation marks in dialogue).

On the other hand, Courtney's selection of a narrow content objective, combined with her assumption about her students' lack of prior experience with Scratch, limited students' sustained engagement with coding. Although content and CT were integrated, students engaged with the coding activity for a single class period. Observation of Courtney's lesson also illustrated the impact of her assumption that students had no prior experience with Scratch. In her interview, Courtney described how she framed the lesson to students: "I'm learning right along with you." However, observation of her lesson revealed that at least two students had prior experience with Scratch, and several other students also worked ahead as Courtney guided them through the activity using a step-by-step instructional video. Courtney's assumption that students would share her own level of experience with Scratch limited opportunities for sustained engagement and more complex applications of coding.

Nonetheless, Courtney's implementation, viewed alongside her lesson plan, demonstrates the *enhance* level of integration (Coenraad et al., 2022). While the *exhibit* level of integration is characterized by the use of CT activities to reinforce prior learning, Courtney's lesson implementation clarified her use of CT integration to build students' disciplinary content knowledge, a key characteristic of the *enhance* level of integration.

Case Study 2: Ana & Emma

Planning

As noted, we present Ana and Emma as a single case because of their collaborative approach to integrating CT in content area instruction and treatment of CT integration as a joint goal. Ana (mathematics) and Emma (CS) attended the PD together and taught the same group of fifth-grade students; all students who attended Ana's mathematics class also attended Emma's standalone CS course. Each teacher developed a three-day plan; Ana and Emma then implemented their plans sequentially.

Ana and Emma's decision to integrate CT in mathematics was obvious given Ana's role as a mathematics teacher. First, Ana provided content instruction in her mathematics class, and Emma subsequently worked with students on a CT application of the content during her CS class. Ana and Emma illustrate well the *exhibit* level of integration; the sequential nature of their lessons reflects the use of CT to reinforce previous learning, an approach that is characteristic of the *exhibit* level (Coenraad et al., 2022).

Ana's plan focused strictly on geometric transformations in the coordinate plane and did not include any explicit CT concepts or vocabulary, demonstrating her reliance on Emma for CT-related instruction. In Emma's CS class, students used Scratch to animate geometric transformations. Thus, Emma leveraged the mathematical content previously taught by Ana. While mathematical concepts appeared in Emma's lesson plan, they functioned as reinforcement of prior learning, rather than integration of CT to advance content learning. The collaboration between the two teachers, therefore, limited the extent to which integration was apparent within *each* of their respective classes. In fact, the teachers' collaboration structure illustrates the separation between CT and content characteristic of the *exhibit* level of integration, where CT and content "do not overlap" (Coenraad et al., 2022, p. 13).

Both Ana and Emma thought that Scratch was an appropriate tool for a geometry project. For Ana, Scratch "seemed to be the program that would facilitate whatever [math] topic I wanted to teach." Emma commented, "I think that Scratch is the foundation – a logical place to begin with computing." In addition, both teachers and their students had significant prior experience with Scratch: "the kids use it a lot." Ana had used Scratch the previous year, and Emma brought over a decade of experience teaching with it.

This case demonstrates integration *across* the two teachers, with mathematics and CT operating as two independent components yet reinforcing one another. The CT activity served as an opportunity for students to demonstrate skills and content learned elsewhere; the mathematics concepts became the "material" for the students' learning of CT. Thus, the case illustrates Coenraad et al.'s (2022) *exhibit* level of integration.

Implementation

Ana's lesson implementation focused on mathematics. Although she mentioned to her students that they would later use the geometry content to complete a Scratch project in Emma's class, this was the only observed connection to CT. Observations of Emma's lesson indicated her expectation that students had a strong understanding of the mathematics content covered in Ana's class, as the CT activity required students to leverage their prior knowledge to explain geometric transformations to someone unfamiliar with the concepts. Although Emma included some mathematics content by displaying visual representations of transformations and asking students to briefly recall their content learning prior to the CT activity, her lesson did not extend students' content knowledge. Indeed, she was observed to make some inaccurate statements when reminding students of their previous content learning (e.g., that only squares or equilateral triangles "remain the same shape" when transformed, whereas by definition all shapes do). This is further evidence that Emma's expertise and role lay in CS, while Ana's lay in mathematics.

Emma emphasized her reliance on Ana to provide the mathematics instruction required for students to complete the CT activity during her interview:

> Some of [the students] were mixing up the concepts of math. … I told Ana, 'Ana, there are kids that don't understand … so either you come to my class, and you teach them that or I'll send them individually [to you].'

In practice, Emma's lesson functioned as an assessment of students' previous content learning, which is characteristic of the *exhibit* level of CT integration (Coenraad et al., 2022). A three-week gap between the end of Ana's lesson and the beginning of Emma's underscored the separate roles of each teacher in their collaborative integration.

Emma circulated during the lesson to assist students as needed, occasionally addressing an individual student's question to the whole class. On at least one occasion, she clarified a point related to content, stating that "the reflection is flipped. When we flip it, it's the other side." However, the majority of her clarifications addressed CT-related project requirements, such as the number of costumes students must include for their sprites. Emma's lesson plan assumed that students had significant prior experience with Scratch and were well-equipped to complete the CT activity, which was confirmed during observations of her class and student interviews.

Both Ana and Emma commented on how much time students spent on the CT activity, with some students working beyond the project requirements because they were so interested in Scratch. While Emma's planned lesson spanned three days, her implementation stretched across five days. Emma's dedicated CS course and the absence of a required curriculum facilitated this opportunity for sustained engagement with coding beyond the multiple days originally included in her plan.

Some students appeared to connect their mathematics knowledge to coding, as recounted in a conversation Emma had with a student. According to Emma,

> [The student said] 'Miss E., remember that I was flipping the shapes? I learned how I can flip a costume and then animate the movement in a different way,' and [I'm] like, 'yeah, there you go. … Everything is shapes [*sic*].'

Similarly, another student explained that the CT activity prompted him to think about how understanding the positive and negative axes in the coordinate plane helped him correctly code the movement of geometric figures on the background of his Scratch project. This student was able to identify how content area knowledge acquired during Ana's earlier mathematics instruction was reflected in the coding activity. At times, Emma and the students made connections between mathematics content and CT. However, the primary role of Emma's CT-integrated lesson was to reinforce, rather than extend, previous learning, which is characteristic of the *exhibit* level of integration (Coenraad et al., 2022).

DISCUSSION

Within- and Across-Course CT Integration

In this work, we presented a PD program that prepares elementary teachers to integrate CT in content area instruction. The two cases described offer contrasting examples of integrating CT in core content in elementary school settings. As noted, there were differences in school context, teaching structure, content area, and level of integration, while the primary similarity was the choice of computing tool introduced during the PD (i.e., Scratch). In both cases, however, we found that participants demonstrated integration of CT and content in their lesson planning and implementation, albeit at different levels of integration (Coenraad et al., 2022; Waterman et al., 2020). In particular, findings indicated that both the context and participants' related instructional decisions during planning and implementation contributed to the level of integration achieved.

The overarching theme that emerged from the two cases centered on the contrast between Courtney's integration of CT *within* her content area class and Emma and Ana's integration *across* their respective content and CS classes. As a generalist classroom teacher, Courtney leveraged the opportunity to integrate CT *within* core content, specifically ELA. Ana and Emma's course structure, on the other hand, in which all students were simultaneously enrolled in both mathematics and CS courses, facilitated an opportunity for them to integrate CT and content *across* their courses. The unique contexts of the two models revealed different affordances and constraints, specifically related to time and prior experience with computing.

Affordances and Constraints

Courtney's CT integration took place *within* her existing class structure, a necessary response to the challenge of finding *time* to integrate CT without the assistance of a dedicated CS teacher or class. This finding is consistent with existing research indicating that instructional time constraints can pose a significant barrier to the integration of CT in elementary schools (Century et al., 2020; Rich & Hu, 2019; Rich et al., 2021). However, Courtney's integration of CT in ELA is consistent with prior research that has shown content area CT integration to be an effective way to overcome time constraints (Waterman et al., 2020), particularly given her perception that ELA afforded more flexibility for CT integration compared to mathematics.

Ana and Emma also integrated CT and content, but the additional time provided by Emma's standalone CS course removed the impetus for Ana to integrate CT in her content instruction to the same extent as Courtney. Although Ana, like Courtney, was required to cover specific content, her reliance on Emma to implement the CT portion of the lesson lessened the pressure to integrate CT activities within strict curricular timelines. Rather, Emma's role as the CS teacher established her responsibility to provide all instruction that included CT.

This finding suggests that CT integration can occur both within core content classes as well as through collaboration of content area and CS teachers. Nonetheless, it is important to note that schools in urban, rural, and other under-resourced communities remain less likely to provide designated CS instruction (Code.org et al., 2022). In order to avoid perpetuating inequities, it is therefore critical to continue examining different models of CT integration within core content. Further, it is important to identify ways in which CT resources could be mapped to content area curricula or standards (e.g., CCSS). Such efforts are currently underway (e.g., California Department of Education, 2021).

Courtney's lack of *prior experience* with computing tools also influenced her decisions about how to integrate CT in her lesson. During the PD, she found Scratch to be most accessible to her as a beginner and mistakenly assumed that students would share her stance as a beginner. On the other hand, we observed during her lesson implementation that Courtney's students demonstrated more proficiency using Scratch than Courtney anticipated. This finding is particularly revealing because it shows that both teacher confidence and predictions about students' success may inform teachers' approaches to CT integration.

Ana and Emma also cited prior experience as an influential factor in their decision to use Scratch in their CT-integrated lesson. Unlike Courtney, however, their rationale hinged on their own and their students' previous use of the tool, rather than its perceived accessibility for beginners. Nonetheless, in both cases, we observed a level of student comfort and experience with Scratch that could have facilitated more complex and deeper engagement with CT than the teachers planned. The teachers' decisions to integrate Scratch based on either prior experience or lack thereof suggests that PD should equip teachers new to CT integration with the confidence to integrate such tools while also preparing teachers with prior experience to integrate varied tools in more complex ways.

Depth of CT Integration

Considering these cases in the context of the frameworks provided by Waterman et al. (2020) and Coenraad et al. (2022), we see Courtney as an example of the *enhance* level of integration and Ana and Emma together as demonstrating the *exhibit* level. As shown in Figure 1, the key difference between these levels is how content area and CT learning are positioned relative to each other. At the *enhance* level, one or more CT activities are added to a content area lesson to "*enhance* the disciplinary concept and provide clear connection to computing concepts that are present, but not central, to the existing lesson" (Waterman et al., 2020, p. 55).

The need to integrate CT *within* content area instruction contributed to Courtney's lesson reaching the *enhance* level of integration (Coenraad et al., 2022). As a generalist classroom teacher, time constraints posed by Courtney's curriculum required her to consider CT integration in a way that did not treat CT as an "add-on." Courtney responded to this challenge by developing and implementing a lesson in which additional tasks both connected to CT and contributed to students' mastery of disciplinary objectives, thereby reaching the *enhance* level of integration.

At the *exhibit* level of integration, lessons "use a CT activity, typically programming, to exhibit knowledge students gained through other means" (Coenraad et al., 2022, p. 13). Coenraad et al. (2022) argues that a lesson at the *exhibit* level does not extend and continue disciplinary learning, but rather "can only serve as an assessment of content understanding" (Coenraad et al., 2022, p. 16). This distinction may be a helpful way of interpreting how mathematics was incorporated to some extent in Emma's CS class. The additional time and flexibility offered by Emma's standalone CS course allowed Emma and Ana to integrate content and CT collaboratively. Additional flexibility created less pressure to move students forward in disciplinary learning and provided time for prolonged student engagement with tasks that integrated CT and mathematics content. Ana's frontloading of mathematics content prior to CT lessons and the choice of a computing tool with which students were already familiar meant that Emma's lesson primarily served as reinforcement of previous learning, aligning with the *exhibit* level of CT integration.

Coenraad et al. (2022) avoids positioning *exhibit* in a hierarchy of integration. Like them, we viewed *exhibit* and *enhance* as different, not better or worse, and analyzed our two cases accordingly. Looking at the different contexts, though, we posit that Courtney's reality is more typical in the United States. Few elementary schools have standalone CS courses or designated CS teachers like Emma (Code.org et al., 2022). Rather, it is more common for elementary teachers to cover multiple subjects and to be new to coding (Code.org et al., 2023). Courtney's case suggests that this context, while posing challenges, can actually support and facilitate the integration of CT in ways that build students' content knowledge.

IMPLICATIONS

Findings from our work have practical implications related to PD and CT integration, as well as conceptual implications related to frameworks that help researchers and PD designers assess the levels of CT integration in teachers' instructional planning and lesson implementations.

Practical Implications for CT Integration and PD Design

Existing research demonstrates that integration in content area instruction is an effective approach to providing CT education to elementary students (Waterman et al., 2020). Yet most integration efforts to date have focused in STEM areas, such as science (e.g., Coenraad et al., 2022; Waterman et al., 2020). Our work contributes to the literature by illustrating a model of integration in ELA. In fact, Courtney identified ELA as the most feasible subject area in which to integrate CT, suggesting that further exploration of CT integration in ELA is warranted. PD designers should help generalist teachers explicitly consider the affordances and constraints of different subject areas, such as flexibility in pacing and time, as teachers approach the planning process. Locating and providing examples of CT integration in ELA could aid teachers in considering how to best integrate CT while meeting curricular demands.

Our findings also reveal the possibilities that are created by the growing repository of existing CT-integrated content area lesson plans, such as the resources available via *CS First* that were introduced in our PD. The alignment of many existing resources with content standards can increase teacher confidence by identifying concrete opportunities to integrate CT while meeting demands posed by required standards and curricula, particularly for teachers without prior computing experience, such as Courtney. Findings from our work suggest that given the wealth of existing resources, PD focused on adapting pre-existing lessons, rather than creating new ones, may be an effective approach to scaling up teacher preparation for CT integration. Thus, PD facilitators should dedicate time to exploring these resources with teachers, especially resources aligned with local curricula.

Our work also suggests that extending the duration of PD beyond lesson implementation may benefit teachers without prior CT experience in particular. After gaining confidence from implementing a CT-integrated lesson, teachers may design lessons that achieve deeper levels of integration. The narrow objectives and limited time that characterized Courtney's lesson, for example, suggest that she may have been "testing out" the idea of integrating CT in her instruction. An iterative PD structure could both improve teachers' CT integration and increase their capacity to sustain integration in future instruction.

We also found that including CS teachers alongside content area partners during PD can facilitate collaboration that contributes to CT integration. However, supporting teachers who work collaboratively requires PD designers to differentiate existing PD models that often focus on preparing content area teachers to independently integrate CT *within* their classes. PD with an intentional focus on both CT and disciplinary content, therefore, is particularly important for CS and content area teacher partners. More specifically, PD should prepare teacher partners to collaborate equitably in *both directions*; in other words, the content area teacher should integrate CT while the CS teacher integrates content. Ensuring that the content area teacher actively participates in CT integration, rather than simply providing content for the CS teacher to integrate in CT tasks, better aligns with the goal that content area teachers provide CT-integrated instruction to reach all students.

Conceptual Implications

In addition to practical implications related to PD and CT integration, findings from this work have implications for expanding Coenraad et al.'s (2022) framework to deepen teachers' efficacy in CT integration. Coenraad et al. utilize their integration framework to assess the level of CT integration in teachers' lesson plans following participation in PD. Like Coenraad et al., we evaluated teachers' lesson plans for CT integration, but we also extended our use of the framework to interpret our observations of participants' lesson implementations. In both cases, we observed that the CT integration evident in teachers' instruction and students' work extended beyond the integration evident in their lesson plans. This finding suggests that restricting an evaluation of CT integration to lesson plans may not provide the most accurate assessment of CT integration as it occurs in practice. Observing lesson implementations can provide additional insight and allow PD facilitators to increase the efficacy of future offerings. Observations of teachers' lesson implementations could also be used to design targeted follow-up PD that supports teachers in reaching deeper levels of CT integration.

Further, findings suggest that the framework itself could be leveraged in the design of initial PD, rather than solely as an assessment mechanism employed following teacher participation in PD programs. Presenting the framework as a tool to guide teachers' thinking and decision-making as they plan CT-integrated lessons, however, would require further discussion about how the *exhibit* level of integration (Coenraad et al., 2022) fits into the hierarchy.

LIMITATIONS

There are two primary limitations reflected in this work. First, the sample only includes three teachers who, together, provided two cases to explore. Although five teachers initially participated in the program, only the three teachers included in the study completed the PD, lesson planning, classroom implementation, and data collection components of the program. The limited sample size and participant attrition was significantly impacted by the COVID-19 pandemic. Second, teachers included in the study volunteered to complete the program and, therefore, already demonstrated motivation to integrate CT in content area instruction. As a result, the two cases do not indicate generalizability of findings. However, our in-depth exploration of each case, particularly given the contrasting contexts, illuminates new possibilities for CT integration models and important areas for further exploration. Our analysis of the ways in which teachers integrated CT with content in an independent school that offered additional opportunities for CS education adds to existing research focused on how to overcome limited opportunities for CS education that are more characteristic of public-school contexts.

CONCLUSION

The integration models illustrated in our two cases provide insights into teacher learning about the integration of CT and content. Future PD should attend to the varied contexts in which teachers integrate CT by including models for CT integration both *within* content area classes and *across* classes. In particular, PD designers should differentiate existing PD activities to more effectively prepare teachers to implement collaborative CT integration in schools with standalone CS courses, while continuing to prepare content area teachers to independently integrate CT in their instruction. Additionally, PD should directly address factors involved in teachers' decision-making processes as they plan and implement CT-integrated lessons. Our findings indicate that teachers' decisions during lesson planning and implementation are influenced by their familiarity with computing tools, as well as the time and flexibility afforded in their specific contexts. PD should therefore be designed to address these various contexts and maximize teacher success in developing and implementing CT-integrated core content area lessons.

Additionally, research that explores whether and how teachers sustain CT integration beyond the lessons planned during PD would provide insight into how to effectively build capacity for ongoing CT integration. Future research that further examines what different levels of CT integration look like, specifically in relation to contextual factors such as content area and teacher comfort level with computing, might also reveal opportunities to use existing frameworks (Coenraad et al., 2022; Waterman et al., 2020) to deepen teachers' CT integration. In particular, future work should consider how the application of existing frameworks can be extended beyond the evaluation of instructional plans to gain a deeper understanding of the levels of CT integration apparent in lesson implementations.

Finally, the ultimate goal of teacher PD is to influence student outcomes; however, this study focuses only on teachers' lesson planning and implementation, recognizing their importance in connecting PD to student learning. Future work should explore student outcomes related to both CT skills and content area knowledge following lesson implementation, which could further inform understandings of effective CT and content integration. Additional exploration of how Coenraad et al.'s (2022) *exhibit* level of integration might fit hierarchically in the framework could provide insights into its application and the extent to which this level of integration contributes to the development of students' CT skills and disciplinary content knowledge. Our ongoing research on the PD program presented in this work will provide opportunities to continue to explore these questions.

ACKNOWLEDGMENTS

This work is funded by a grant from the National Science Foundation, Award# 1923483.

REFERENCES

Adler, R. F., & Kim, H. (2018). Enhancing future K–8 teachers' computational thinking skills through modeling and simulations. *Education and Information Technologies*, *23*(4), 1501–1514.

Ardito, G., Mosley, P., & Scollins, L. (2014). WE, ROBOT: Using robotics to promote collaborative and mathematics learning in a middle school classroom. *Middle Grades Research Journal*, *9*(3), 73–88.

Aslan, U., Horn, M., & Wilensky, U. (2024). Why are some students "not into" computational thinking activities embedded within high school science units? Key takeaways from a microethnographic discourse analysis study. *Science Education*, *108*(3), 929–956. https://doi.org/10.1002/sce.21850

Bers, M. U., Govind, M., & Relkin, E. (2022). Coding as another language: Computational thinking, robotics and literacy in first and second grade. In A. Ottenbreit-Leftwich & A. Yadav (Eds.), *Computational thinking in PreK–5: Empirical evidence for integration and future directions* (pp. 30–38). Association for Computing Machinery. https://doi.org/10.1145/3507951.3519285

Burke, Q., & Kafai, Y. B. (2010, June). Programming & storytelling: Opportunities for learning about coding & composition. In *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 348–351). Association for Computing Machinery. https://doi.org/10.1145/1810543.1810611

Burke, Q., & Kafai, Y. B. (2012, February). The writers' workshop for youth programmers: Digital storytelling with Scratch in middle school classrooms. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education* (pp. 433–438). Association for Computing Machinery. https://doi.org/10.1145/2157136.2157264

Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with Scratch: An experiment with 6th grade students. In Conole, G., Klobučar, T., Rensing, C., Konert, J., & Lavoué, E. (Eds.) *Design for teaching and learning in a networked world*. EC-TEL 2015 (pp. 17–27). https://doi.org/10.1007/978-3-319-24258-3_2

California Department of Education. (2021). *Computer science standards for California public schools*. https://www.cde.ca.gov/be/st/ss/documents/csstandards.pdf

Century, J., Ferris, K. A., & Zuo, H. (2020). Finding time for computer science in the elementary school day: A quasi-experimental study of a transdisciplinary problem-based learning approach. *International Journal of STEM Education*, *7*, Article 20. https://doi.org/10.1186/s40594-020-00218-3

Cheryan, S., Master, A., & Meltzoff, A. N. (2015). Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in Psychology, 6*, Article 49. https://doi.org/10.3389/fpsyg.2015.00049

Ching, Y. H., Hsu, Y. C., & Baldwin, S. (2018). Developing computational thinking with educational technologies for young learners. *TechTrends*, *62*(6), 563–573. https://doi.org/10.1007/s11528-018-0292-7

Code.org, CSTA, & ECEP Alliance. (2022). *2022 State of computer science education Understanding our national imperative*. https://code.org/assets/advocacy/stateofcs/2022_state_of_cs.pdf

Code.org, CSTA, & ECEP Alliance. (2023). *2023 State of computer science education Understanding our national imperative*. https://code.org/assets/advocacy/stateofcs/2023_state_of_cs.pdf

Coenraad, M., Cabrera, L., Killen, H., Plane, J., & Ketelhut, D. J. (2022). Computational thinking integration in elementary teachers' science lesson plans. In Ottenbright-Leftwich, A. & Yadav, A. (Eds.), *Computational thinking in K–5: Empirical evidence for integration and future directions* (pp. 11–18). Association for Computing Machinery. https://doi.org/10.1145/3507951.3519283

DeLyser, L. A., Goode, J., Guzdial, M., Kafai, Y., & Yadav, A. (2018). *Priming the computer science teacher pump: Integrating computer science education into schools of education*. CSforAll. https://www.csfored.org/report2018

Desimone, L. M. (2009). Improving impact studies of teachers' professional development: Toward better conceptualizations and measures. *Educational Researcher*, *38*(3), 181–199.

Education Development Center, Inc. (2024). *The CT integration framework*. https://ctframework.edc.org/framework/

Elo, S., & Kyngäs, H. (2008). The qualitative content analysis process. *Journal of Advanced Nursing, 62*(1), 107–115. https://doi.org/10.1111/j.1365-2648.2007.04569.x

Feng, S., & Yang, D. (2022). Teachers' perceived value, challenges and advice for implementing computational thinking in elementary classrooms. *Journal of Technology and Teacher Education, 30*(3), 293–320.

Fields, D. A., Kafai, Y. B., Strommer, A., Wolf, E., & Seiner, B. (2014). Interactive storytelling for promoting creative expression in media and coding in youth online collaboratives in Scratch. In G. Futschek & C. Kynigos (Eds.), *Proceedings of Constructionism* (pp. 19–23). Austrian Computer Society.

Fletcher, C. L., Cox, B., DeLyser, L., & Ko, A. J. (2021). Applying CAPE to assess equitable CS education. In C. Gardner-McCune, S. Grady, Y. Jimenez, J. Ryoo, R. Santo, & J. Payton (Eds.), *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)* (pp. 250–251). IEEE. https://doi.org/10.1109/RESPECT51740.2021.9620690

Gay, G. (2018). *Culturally responsive teaching: Theory, research, and practice* (3rd ed.). Teachers College Press.

Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends*, *60*(5), 510–516. https://doi.org/10.1007/s11528-016-0098-4

Hershkowitz, H., Bain, C., Kelter, J., Peel, A., Wu, S., Horn, M. S., & Wilensky, U. (2023). Contribution of computational thinking to STEM education: High school teachers' perceptions after a professional development program. *Journal of Computers in Mathematics and Science Teaching*, *42*(1), 35–65.

International Society for Technology in Education (ISTE), & Computer Science Teachers Association (CSTA). (2011). *Operational definition of computational thinking for K–12 education*. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf

Iskrenovic-Momcilovic, O. (2020). Improving geometry teaching with Scratch. *International Electronic Journal of Mathematics Education, 15*(2), Article em0582. https://doi.org/10.29333/iejme/7807

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, *82*(1), 263–279. https://www.learntechlib.org/p/201785

Jocius, R., Joswick, C., Albert, J., Joshi, D., & Blanton, M. (2023). Towards pedagogical content knowledge learning trajectories: Tracing elementary teachers' infusion of computational thinking. *Professional Development in Education*, 1–20. https://doi.org/10.1080/19415257.2023.2228813

Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, *73*(26), 26–39. https://doi.org/10.1016/j.compedu.2013.12.010

Ladson-Billings, G. (1995). Toward a theory of culturally relevant pedagogy. *American Educational Research Journal*, *32*(3), 465–491.

Lambic, D. (2011). Presenting practical application of mathematics by the use of programming software with easily available visual components. *Teaching Mathematics and its Applications: An International Journal of the IMA*, *30*(1), 10–18. https://doi.org/10.1093/teamat/hrq014

La Paglia, L., Cascia, L., Francomano, M. M., & Barbera, L. (2017). Educational robotics to improve mathematical and metacognitive skills. *Annual Review of CyberTherapy and Telemedicine*, *15*(14), 70–75.

Margulieux, L. E., Enderle, P., Clarke, P. J., King, N., Sullivan, C., Zoss, M., & Many, J. (2022). Integrating computing into preservice teacher preparation programs across the core: Language, mathematics, and science. *Journal of Computer Science Integration*, *5*(1), 1–16. https://doi.org/10.26716/jcsi.2022.11.15.35

Master, A., Meltzoff, A. N., & Cheryan, S. (2021). Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. In *Proceedings of the National Academy of Sciences, 118*(48), Article e2100030118. https://doi.org/10.1073/pnas.2100030118

McGinnis, J. R., Hestness, E., Mills, K., Ketelhut, D. J., Cabrera, L., & Jeong, H. (2020). Preservice science teachers' beliefs about computational thinking following a curricular module within an elementary science methods course. *Contemporary Issues in Technology and Teacher Education, 20*(1). https://citejournal.org/volume-20/issue-1-20/science/preservice-science-teachers-beliefs-about-computational-thinking-following-a-curricular-module-within-an-elementary-science-methods-course

Merriam, S. B. (1998). *Qualitative research and case study applications in education.* Jossey-Bass.

Merriam, S. B. (2009). *Qualitative research: A guide to design and implementation*. Jossey-Bass.

Mills, K., Coenraad, M., Ruiz, P., Burke, Q., & Weisgrau, J. (2021, December). *Computational thinking for an inclusive world: A resource for educators to learn and lead*. Digital Promise. https://doi.org/10.51388/20.500.12265/138

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*, Article 104530. https://doi.org/10.1016/j.compedu.2022.104530

Mouza, C., Yadav, A., & Leftwich, A. (2021). Preparing pre-service teachers to teach computer science. In C. Mouza, A. Yadav, & A. Ottenbreit-Leftwich (Eds), *Preparing teachers to teach computer science: Models, practices and policies* (pp. xiii–xxii). Information Age Publishing.

Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y. & Pollock L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*, *33*(3), 61–76. https://doi.org/10.14742/ajet.3521

Mulhall, A. (2003). In the field: Notes on observation in qualitative research. *Journal of Advanced Nursing*, *41*(3), 306–313. https://doi.org/10.1046/j.1365-2648.2003.02514.x

National Governors Association Center for Best Practices, & Council of Chief State School Officers. (2010). *Common Core State Standards*. https://corestandards.org

NGSS Lead States. (2013). *Next generation science standards: For states, by states*. The National Academies Press. https://www.nextgenscience.org/

Pektas, E., & Sullivan, F. (2021). Storytelling through programming in Scratch: Interdisciplinary integration in the elementary English language arts classroom. In C. K. Looi, B. Wadhwa, V. Dagiene, P. Seow, Y. H. Kee, & L. K. Wu (Eds.), *Proceedings of the Fifth Asia Pacific Society for Computers in Education International Conference on Computational Thinking and STEM Education* (pp. 38–42). Asia Pacific Society for Computers in Education.

Qian, Y., Hambrusch, S., Yadav, A., & Gretter, S. (2018). Who needs what: Recommendations for designing effective online professional development for computer science teachers. *Journal of Research on Technology in Education, 50*(2), 164–181. http://eric.ed.gov/?id=EJ1174427

Repenning, A., Webb, D. C., Koh, K. H., Nickerson, H., Miller, S. B., Brand, C., Her Many Horses, I., Basawapatana, A., Gluck, F., Grover, R., Gutierrez, K., & Repenning, N. (2015). Scalable game design: A strategy to bring systemic computer science education to schools through game design and simulation creation. *ACM Transactions on Computing Education, 15*(2), 1–31.

Rich, P. J., & Hu, H. (2019). Surveying the landscape: Statewide data on K–12 CS education implementation. *Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT),* (pp. 1–8), IEEE.

Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education, 168*, Article 104196.

Ryoo, J. J. (2019). Pedagogy that supports computer science for *all. ACM Transactions on Computing Education, 19*(4), 1–23. https://doi.org/10.1145/3322210

Sadik, O., Ottenbreit-Leftwich, A., & Brush, T. (2020). Secondary computer science teachers' pedagogical needs. *International Journal of Computer Science Education in Schools*, *4*(1), 33–52.

Saldaña, J. (2021). *The coding manual for qualitative researchers* (4th ed.). SAGE Publications.

Shehzad, U., Clarke-Midura, J. E., Beck, K., Shumway, J. F., & Recker, M. M. (2023, April). Rethinking integrated computer science instruction: A cross-context and expansive approach in elementary classrooms. Paper presented at the *Annual Meeting of the American Educational Research Association.* https://digitalcommons.usu.edu/eled_support_pubs/6

Sjöberg, C., Nouri, J., Sjöberg, R., Norén, E., & Zhang, L. (2018). Teaching and learning mathematics in primary school through Scratch. In L. G. Chova, A. Lopez Martinez, & C. Torres (Eds.), *EDULEARN18 Proceedings, 10th International Conference on Education and New Learning Technologies* (pp. 5625–5632). IATED. https://doi.org/10.21125/edulearn.2018.1359

Smith, S., & Burrow, L. E. (2016). Programming multimedia stories in Scratch to integrate computational thinking and writing with elementary students. *Journal of Mathematics Education*, *9*(2), 119–131.

SocioCultural Research Consultants (2021). Dedoose Version 9.0.17. SocioCultural Research Consultants, LLC.

Taylor, M., Harlow, A., & Forret, M. (2010). Using a computer programming environment and an interactive whiteboard to investigate some mathematical thinking. *Procedia - Social and Behavioral Sciences*, *8*, 561–570. https://doi.org/10.1016/j.sbspro.2010.12.078

Vivian, R., & Falkner, K. (2019). Identifying teachers' technological pedagogical content knowledge for computer science in the primary years. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 147–155).

Waterman, K. P., Goldsmith, L., & Pasquale, M. (2020). Integrating computational thinking into elementary science curriculum: An examination of activities that support students' computational thinking in the service of disciplinary learning. *Journal of Science Education and Technology*, *29*(1), 53–64. https://doi.org/10.1007/s10956-019-09801-y

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2015). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Whyte, R., Ainsworth, S., & Medwell, J. (2019). Designing for integrated K–5 computing and literacy through story-making activities. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (pp. 167–175). Association for Computing Machinery. https://doi.org/10.1145/3291279.3339425

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33. https://doi.org/10.1145/1118178.1118215

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K–12 classrooms. *TechTrends*, *60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yin, R. (2017). *Case study research and applications* (7th ed.). SAGE Publications.