

The Design of Integrated Computer Science Instruction in the Context of Teacher Professional Development

Abstract

In this work, we present a professional development (PD) program that seeks to support elementary teachers as they integrate computer science (CS) with disciplinary content and culturally responsive pedagogy (CRP) to create inclusive environments that engage all students with computing. Using semi-structured interviews with 17 participants, we subsequently examine the content, technology tools, and CRP strategies that teachers perceived as represented in lessons designed during their participation in PD. Findings indicated that teachers integrated CS tools primarily with literacy and utilized CRP strategies commonly cited as instructional best practices (e.g., differentiation). Results have implications for future PD as well as research that seeks to support teacher learning about CS-integrated instruction.

Keywords: professional development, computer science, culturally-responsive pedagogy

Purpose and Perspectives

Knowledge from computer science (CS) has become essential for navigating the world around us (Wing, 2006). As a result, there has been increased attention on introducing CS to students from the early grades. Although substantial progress has been made in the last decade to bring CS education to schools, it continues to remain one of the most segregated academic subjects in terms of race and gender (Margolis et al., 2017). One way to ensure equitable access to CS in the early grades is by integrating CS to other core subjects taught in school (Waterman et al., 2020). Such approaches not only help broaden participation in CS but also help students connect CS to other disciplines, thus illustrating the relevance and significance of CS in the world (Falkner & Sheard, 2019). However, there are a number of challenges in engaging elementary students with CS. As Du and Meier (2023) point out, integrating CS with content is a complex process and cannot be accomplished by trying to “shoehorn” the new content into already overloaded curricula.

One way to address this challenge is through research-based professional development (PD) that simultaneously builds teacher content and pedagogical knowledge, including ways in which CS can support disciplinary understandings in math, literacy, etc. (Authors, 2022). Such programs should also encourage integration of CS with content in culturally relevant ways (Gay, 2018). Specifically, helping teachers integrate knowledge relevant to student identities and communities with CS is critical for creating inclusive environments that engage all students (Authors, 2021). To accomplish this goal, teachers could benefit from hands-on learning experiences, exploration of new CS tools, and access to sample curricular materials which allows them to integrate new

content into current lesson plans using culturally relevant pedagogy (CRP) (Du & Meier, 2023). In this work, we present one such PD program which seeks to support teachers as they integrate CS with content and CRP in K-5 classrooms. Specifically, we describe the PD program and investigate the following question: *What content, technology tools, and CRP strategies introduced during the PD do teachers perceive as represented in their lesson plans?*

Description of PD Program

This work is situated in a larger effort focused on improving the teaching of CS in elementary grades (K-5) through research-based equity-focused PD (e.g., Authors, 2022; Desimone, 2009). It is structured around two components – a face-to-face week-long Summer Institute and follow-up classroom support (Authors, 2015; 2022). The Summer Institute focuses on preparing teachers to integrate CS principles into existing content area instruction while utilizing CRP practices known to help broaden participation in CS (e.g., Authors, 2021).

The design of the institute is grounded in the framework of technological pedagogical content knowledge (TPACK) which describes the knowledge needed for effective integration of technology in teaching (Mishra & Koehler, 2006). It includes explicit attention to CS resources and tools (technology), CS principles in the context of content-area instruction (content), and CRP (pedagogy). Specifically, three major themes related to CRP and examples of accompanying strategies were presented during the PD, (Madkins et al., 2020; see Figure 1). Teachers culminate their participation in the institute by designing a lesson which integrates CS with disciplinary content and CRP in K-5 instruction. Table 1 presents the design of the Institute.

**** Insert Table 1 and Figure 1 here ****

Methods

Participants

Participants included 17 K-5 teachers in a Mid-Atlantic state who attended the Summer Institute. Teachers were recruited through listservs and other school and local contacts [e.g., Computer Science Teachers' Association (CSTA) local chapter]. All teachers voluntarily participated and were compensated for their time. Table 2 provides participants' demographics.

**** Insert Table 2 here ****

Data Collection & Analysis

Qualitative data were collected from two sources during the Summer Institute: (a) PD materials, teacher artifacts, and observations of PD sessions during the Summer Institute; and (b) teacher interviews (N=17). The semi-structured interview protocol included 12 questions related to the

PD program, the CRP dimension of the PD, lesson planning integration, plans for implementation, and needs for support. All participants were invited to participate via email and offered an additional small stipend to do so. Six interviewers from an external evaluation team conducted interviews concurrently on the last day of the Institute. Interviews were recorded, transcribed, and analyzed qualitatively.

First, we examined interviews with an eye towards the tenets of our PD: content, technology tools & resources, and CRP. Structured coding was used to identify disciplinary content as well as CS concepts (CSTA, 2017) and CS tools. For questions focusing explicitly on CRP, qualitative analysis focused on the specific strategies introduced during the PD. Structured coding based on the three major themes presented during the CRP PD sessions was used to analyze references to CRP in teacher interviews. Table 3 presents the coding scheme.

**** Insert Table 3 here ****

Observation data and various artifacts (e.g., lesson plans) collected during the Summer Institute were used to triangulate interview data and confirm or dismiss emergent themes. Of the 17 teachers interviewed, 15 submitted a lesson plan at the end of the Institute, and these documents were especially fruitful for triangulation.

Results

What content, technology tools, and CRP strategies introduced during the PD do teachers perceive as represented in their lesson plans?

We first present a summary of teacher responses based on interview data, then provide concrete examples to illustrate various lesson design choices and the teacher decision making that lay behind them.

Disciplinary Content

Table 4 presents the disciplinary content represented in teachers' work. Some disciplinary foci were very clear; for instance, several teachers integrated CS into reading lessons on particular books or stories. One math teacher described using an Unplugged strategy to teach a lesson about classifying shapes. Other lessons were interdisciplinary. For example, a teacher described her lesson on robots as incorporating one ELA standard and one math standard.

**** Insert Table 4 here ****

Choosing a content area for integration was a strategic process. Some teachers looked for a match between grade-level content standards and what the CS tools could accomplish. For example, one participant described the decision to integrate resources from CS First into ELA:

“It felt like what we wanted to teach, in terms of characters, how to create characters. It goes hand in hand with the grade level, third grade. So because that's the point where they learn about character setting and stuff like that.”

Many of the participants were required to teach with prescribed curricula. In these cases, teachers sought synergies between their curricular requirements and what they had learned in the PD. For example, two teachers (working as a team) revised a lesson from their school's scripted math curriculum by adding a “ticket in the door.” As one described the rationale:

“we pulled the curriculum and looked for an application problem on Scratch[Lab]. We modified, remixed the Scratch...So the problems, the application problems matched what our curriculum said but it's on Scratch. ...We can use this with our class and we're not breaking our curriculum. We are just adding more of those computer science skills.”

Computer Science Tools & Concepts

As shown in Table 5, the most frequent CS tool described as part of the lessons was Scratch/Scratch Jr., followed by CS First and the most frequently represented concepts were those of algorithms and programming. This preference reflects the emphasis of the PD; programming in Scratch was “what we did most of the week,” as one teacher said. On the other hand, in one school students already had some exposure to Scratch., which teachers planned to extend. Another rationale for Scratch was that it was free and accessible with the technology already in place:

“In the classroom, they all have a computer. So this is something that I can assign to them to do in the class and we can kind of do it together. So that was kind of my thought process [for choosing Scratch]”

As teachers discussed CS First, they often focused on how it supported younger learners. The program provides the scaffolding of audio directions for non-readers or emergent readers. CS First's structure was also seen as helpful: “a little more concrete for students.”

Finally, many teachers found Unplugged activities especially engaging. Four teachers incorporated Unplugged activity into their lessons. In one lesson, a *Let's Go Code* game was used to introduce students to algorithms before beginning CS First. In another lesson, students sorted polygons based on attributes. Here, a teacher describes how that activity simultaneously supported math and CS content:

“Then we have a discussion about, how did you sort them, why'd you sort them that way, what is an attribute? So as far as connecting that to CS, I found one of the standards was about sorting and displaying data and talking about why you've sorted it that way.”

**** Insert Table 5 here ****

CRP Integration

When asked how they integrated CRP into their lessons, teachers identified strategies that closely aligned with those presented during the introductory CRP session (see Table 6). Teachers addressed students' unique identities by infusing opportunities to draw on their personal and cultural background and lived experiences. In some cases, teachers shared how they employed a culturally responsive lens during the planning process itself. One teacher described featuring community assets: “what are things that these kids see in the neighborhood often?”

This theme also included using curricular materials that reflected students' identities and interests. One teacher recounted a time when one of her students who was Black read a story about Bojangles:

“He loved it and he was like... 'I didn't know that Black men could be dancers. Like this kind of dancer, not just hip-hop and music videos. I want to be a tap dancer.' And he starts doing his own research and investigation. . . I think that's the heart of CRP, right? And him seeing someone that... [represents] the groups that his identity is in, males, Black male. His love for dancing, his love for rap and seeing that this is a thing, and people can do it, and you can become great.”

In her lesson, this teacher incorporated this book and then used the motion sensing feature in Scratch to have students make a sprite “tap dance.”

Teachers also described how they ensured that their CS-integrated plans would be accessible to all students by considering students' individual needs. Some talked about differentiating curriculum, instruction or assessment or using universal design in the classroom. Teachers also saw how CS activities could support the development of growth mindset:

“Every day we do a debrief. 'What were you excited about or happy that you did today and what was a challenge and what problems did you solve today? ...So being aware that you might not be there yet, but you will get there.’”

The integration of student agency was evident as teachers described instructional practices. Many purposely built collaboration into their lessons – either the specific “paired programming / ‘driver’ and ‘navigator’” structure taught during the workshop or some other kind of grouping.

Other teachers emphasized student-centered learning, student choice, or reflection in their lessons.

**** Insert Table 6 here ****

Limitations

This study draws primarily on teacher self-reports and descriptions of their instructional planning and decision making. Future steps in this research should include: (1) more detailed content analysis of the lesson plan artifacts and (2) investigation of implementation experiences, including classroom observations, student responses, and teacher reflections.

Significance

This study summarizes the instructional design choices of a group of elementary teachers developing CS-integrated and culturally relevant lesson plans. Findings indicated that ELA was the most frequent discipline integrated with CS while Scratch/Scratch Jr. and CS First were seen as the most logistically and educationally accessible tools for teachers and students. Finally, in terms of CRP, teachers were more likely to integrate CRP strategies that are commonly cited as instructional best practices (e.g., differentiated instruction). Results yield implications for future PD as well as research which seeks to support teacher learning about CS – and ultimately promote early and equitable access to the field for diverse learners.

References

Authors, 2015; 2021; 2022

Computer Science Teachers Association (2017). *CSTA K-12 Computer Science Standards, Revised 2017*. Retrieved from <https://csteachers.org/k12standards/>.

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

Du, X.X., & Meier, E.B. (2023). Innovating Pedagogical Practices through Professional Development in Computer Science Education. *Journal of Computer Science Research*. 5(3): 46-56. DOI: <https://doi.org/10.30564/jcsr.v5i3.5757>

Falkner, K., & Sheard, J. (2019). Pedagogic approaches. In S. Fincher and A. V Robins (Eds), *The Cambridge handbook of computing education research* (pp. 445-480). Cambridge University Press.

Gay, G. (2018). *Culturally responsive teaching: theory, research, and practice*. Teachers College Press.

Jocius, R., Joshwick, 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*, DOI: 10.1080/19415257.2023.2228813

Madkins, T. C., Howard, N. R., & Freed, N. (2020). Engaging equity pedagogies in computer science learning environments. *Journal of Computer Science Integration*, 3(2), 1-27. <https://doi.org/10.26716/jcsi.2020.03.2.1>

Margolis, J., Estrella, R., Goode, J., Jellison Holme, J., & Nao, K. (2017). *Stuck in the shallow end: education, race, and computing*. MIT Press.

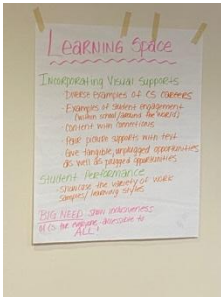
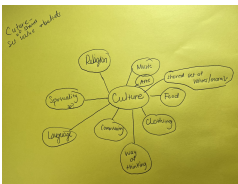
Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, 108 (6), 1017-1054

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>

Wing, J., 2006. Computational thinking. *Communications of the ACM*, 49 (3), 33–35. doi:10.1145/1118178.1118215.

Table 1

Summer Institute PD Content

	Content		Technology	CRP
	<i>CS Content</i>	<i>Disciplinary Content</i>		
Day 1	Introduction to CS & Programming: Programming a trip to Mars through CS Unplugged* Programming in Scratch**: movement & drawing with sequences, repetition & conditionals Program creativity (events, sequencing/reacting to user and keyboard input)	Literacy: Programming in Scratch for storytelling with conversations & animation with different costumes	Scratch  	Introduction to CRP: concept of culture; Importance of CRP in CS; Central themes of CRP: unique identities, individual needs, and student agency
Day 2	Algorithms with CS Unplugged Continue Programming in Scratch Program interactivity (events, sequencing/reacting to user and keyboard input) Coding with music	Introduction to CS First*** Curriculum Disciplinary Content Integration: Lesson planning for integration of CS with disciplinary content and assessment	Advanced Scratch CS First exercises Tunepad**** 	CRP strategies 
Day 3	From algorithms to programming: Introduction to programming with CS robotics, including Micro:bits, beebots, makey-makey*****.	Disciplinary Content Integration: Lesson planning for integration of CS with disciplinary content and assessment	Robotics: beebots, Ozobots, Edison bots. Makey-makey Microbits 	Integration of CRP into lesson designs 

Day 4	Impacts of Computing: AI & Ethics and Cybersecurity	Integration of CSTA standards with disciplinary content.	Working with Data AI tools: face sensing in Scratch & machine learning for kids Cybersecurity online games and resources	Integration of CRP into lesson designs
-------	---	--	--	--

(<https://machinelearningforkids.co.uk/#!/worksheets>)

CS Unplugged*: a collection of free teaching material that teaches Computer Science without technology

Scratch**: object oriented visual programming tool

CS First***: An introductory CS curriculum based on Scratch

Tunepad****: Free online platform for creating music with programming.

Robotics*****: Tools that support CS teaching using physical materials

Table 2*PD Participant Demographics*

Subcode	<i>N</i> = 16 ¹	
	<i>n</i>	%
<i>Gender</i>		
Female	13	81.25
Male	3	18.75
<i>Race</i>		
White	9	56.25
Black/African American	5	31.25
Multiracial	1	6.25
Prefer not to answer	1	6.25
<i>Hispanic or Latino</i>		
Yes	14	87.5
No	1	6.25
Prefer not to answer	1	6.25
<i>Grade Level Taught</i>		
K - 5	16	100.0
<i>Years of Teaching Experience</i>		
1 year	1	6.25
2 - 3 years	2	12.5
4 - 5 years	2	12.5
6 - 10 years	2	12.5
11 - 15 years	3	18.75
16+	6	37.5
<i>Primary Teaching Duties</i>		
All subjects (elementary general or spec. education)	12	75.0
Math	1	6.25
English/social studies/humanities	1	6.25
Other	1	6.25
Not currently teaching	1	6.25
<i>Teaching Certification(s)</i> ²		
Elementary Education	15	93.75
Math	1	6.25
English/social studies/humanities	1	6.25
Other	5	31.25

¹ One interview participant did not complete the demographic pre-survey² For this item, teachers could select more than one response, thus numbers/percentages do not sum to 100.

Table 3*Round 1 Codes*

Codes	Subcodes
Disciplinary content	English language arts Math Science/STEM Social studies Social-emotional learning Other
Computer science tool	Scratch CS First Micro:bits Other robots (e.g., Bee Bots, Ozobots) Unplugged strategies
Computer science concept	Algorithms & Programming Computing Systems Data & Analysis Impacts of Computing Networks & the Internet
Culturally relevant pedagogy strategy	Unique identities Individual needs Student agency

Table 4*Disciplinary Content Represented in Teachers' Work*

Subcode	Application (n=17) ³	
	<i>n</i>	%
English language arts (ELA)	11	64.7
Math	5	29.4
Science/STEM	2	11.8
Social studies	0	0.0
Social-emotional learning (SEL)	2	11.8
Other	1	5.9

Table 5*Technology Tools and Resources Represented in Teachers' Work*

Subcode	Application (n=17) ⁴	
	<i>n</i>	%
<i>Tools</i>		
Scratch/Scratch Jr.	9	52.9
CS First	5	29.4
Micro:bits	1	5.9
Other robots (e.g., Bee Bots, Ozobots)	2	11.8
Unplugged strategies	4	23.5
<i>Concepts⁵</i>		
Algorithms & Programming	14	82.4
Computing Systems	0	0.0
Data & Analysis	3	17.6
Impacts of Computing	1	5.9
Networks & the Internet	0	0.0

³ Four plans were interdisciplinary, thus coded in more than one subcode.

⁴ Some teachers included more than one computer science tool

⁵ One teacher referred to both Algorithms & Programming and Data & Analysis; one teacher referred to both Algorithms & Programming and Impacts of Computing. One teacher interview only cited CS tools but the CS concepts could not be determined.

Table 6

CRP Strategie

Subcode	Application (n=17) ⁶	
	<i>n</i>	%
Unique Identities	5	29.4
Individual Needs	9	52.9
Student Agency	9	52.9

⁶ Some teachers described more than one CRP strategy in their interviews

Figure 1

CRP Themes and Strategies Presentation Slides

