Journal of Pedagogical Research
Volume7, Issue 2, 2023
https://doi.org/10.33902/JPR.202318599

Research Article

Developing preservice teachers' intuitions about computational thinking in a mathematics and science methods course

Peter F. Moon[1], Joshua Himmelsbach[2], David Weintrop[3] and Janet Walkoe[4]

[1]*University of Maryland, United States (ORCID: 0000-0003-0297-3968)*
[2]*University of Maryland, United States (ORCID: 0009-0003-4324-8575)*
[3]*University of Maryland, United States (ORCID: 0000-0002-3009-3899)*
[4]*University of Maryland, United States (ORCID: 0000-0002-5675-9767)*

Computational thinking (CT) has the potential to enhance learning when integrated into mathematical classroom activities. Teachers are being asked to include CT concepts in their core disciplines; however, there is an open question as to how best to equip teachers to integrate CT into their practice. Oftentimes teacher candidates enter math and science methods courses with emerging ideas of what CT might be but little formal experience with the construct (Yadav et al., 2014). Relatively little is understood about the most effective ways to support candidates' understanding of CT, and how to support them in integrating CT into disciplinary instruction. In this paper, we describe a novel method of introducing teacher candidates to CT through a five-lesson module within the context of an existing pre-service teacher math and science methods course. We use an Experience First, Formalize Later format inspired by Stats Medic (2018) to help develop teacher candidates' intuitions around CT primarily through firsthand experience and the roles it can play in their math and science classrooms. This paper presents the instructional materials for this innovative approach for integrating CT into a pre-service mathematics and science methods course. We will also present data from teaching these materials with a cohort of 14 teacher candidates. Collectively, this work contributes a novel strategy for integrating CT into pre-service methods courses and contributes to our understanding of the relationship between CT and the existing disciplines of K-12 math and science, especially as seen by teacher candidates entering the profession.

Keywords: Computational thinking; Integrated computational thinking; Pre-service teacher preparation; Computational thinking in preservice teacher education

Article History: Submitted 20 October 2022; Revised 9 March 2023; Published online 10 June 2023

1. Introduction

In her influential paper, Wing (2006) described CT as "a fundamental skill for everyone," (p. 33), citing applications to statistics, biology, and a myriad of other disciplines and recommending that we teach CT to all children. Further work has argued that CT can be supportive of learning across the K-12 level, particularly in STEM disciplines (Weintrop et al., 2016). However, while efforts to expose teachers to what CT is and how to use it in the classroom exist (e.g., Yadav et al., 2014), there remain significant open questions about how best to prepare future teachers to utilize CT in

Address of Corresponding Author

Peter F. Moon, College Park, MD 20742, University of Maryland, United States.

pmoon@umd.edu

How to cite: Moon, P. F., Himmelsbach, J., Weintrop, D., & Walkoe, J. (2023). Developing preservice teachers' intuitions about computational thinking in a mathematics and science methods course. *Journal of Pedagogical Research, 7*(2), 5-20. https://doi.org/10.33902/JPR.202318599

their classrooms. As CT is a burgeoning field, there are also few efforts to study the effects of CT instruction in teacher education and teacher knowledge of CT.

1.1. Computational Thinking is Potentially Supportive for Learning Mathematics

CT-integrated mathematics shows promise as an instructional model. The potential of grounding mathematics instruction in concepts and practices from computing has a long history dating back to early work on the Logo programming language helping students better understand Geometry (Papert, 1972; 1980). Since this early foundational work, many projects have investigated ways programming, computers, and now CT, can serve as a productive context for supporting mathematical learning. Some projects continued to use the Turtle Geometry programming style started by Logo (Abelson & diSessa, 1986; Wilensky, 1995), while others explored other applications, like SimCalc, which supports algebraic functions, calculus, and Cartesian coordinate graphing (Kaput et al., 2002; Noss & Hoyles, 1996; Roschelle et al., 2000). A recent example is the Sphero.Math project (Weintrop et al., 2022), which has students design and implement programs as part of CT-integrated math lessons in fourth-grade classrooms. This work indicates that well-designed activities can link mathematics and CT in a "mutually-supportive" fashion (Bih et al., 2020, p. 1391).

However, research has revealed how attempts to integrate CT into disciplinary contexts can fail to achieve the desired "mutual supportiveness." For example, in the Sphero.Math project one of the activities asked students to program their Sphero to follow a "prime number path" through a grid comprised of prime and composite numbers. When the lesson was implemented, students did the math (found the prime numbers) first, followed by writing the code for the path which was perceived as a distinct activity from the mathematics by the learners (Bih et al., 2020). This work demonstrates less productive approaches for CT integration: specifically, designing activities that tangentially involve math without integrating the CT task into the mathematical goals of the lesson.

A more productive example of CT in math integration can be seen with Bootstrap: Algebra, "an early-programming curriculum that is designed to teach key algebra topics as students build their own video games," (Schanzer et al., 2015, p. 1). Throughout this curriculum, programming concepts are aligned with math concepts in the broader context of creating a video game. For example, in one lesson, the desired game feature is to "determine when game elements are off-screen," (Schanzer et al., 2015, p. 2). To accomplish this, students must use Boolean logic and, to complete the conditional statements, must define proper algebraic inequalities to make the game behave correctly. By integrating the programming goal of the video game feature, students gain a practical motivation to master a concept in algebra. The programming task, then, encourages students to master this algebraic concept not in the abstract but for a desired goal.

1.2. Researcher and Teacher Understanding of Computational Thinking

One major goal of previous CT research has been to develop and understand conceptualizations of CT, often with the goal of presenting CT to teachers organized towards integration into the classroom. Part of this work has been the creation of academic frameworks to describe CT and its domains. Shute and colleagues (2017) reviewed the state of the CT literature and identified six central facets:

1. Decomposition - dissecting a complex problem into manageable parts
2. Abstraction - extracting the essence of a system
3. Algorithms - logical and ordered instructions for a solution to a problem
4. Debugging - detecting, identifying, and fixing errors with a solution
5. Iterations - refining a solution through a repeated design process
6. Generalization - transferring CT skills to many kinds of problems to solve them efficiently

These facets are comparable to the key CT components listed by Dong and colleagues (2019) in a second operationalization of CT:

1. Pattern Recognition - identifying patterns, trends, and regularities in data, processes, or problems
2. Abstraction [see above]
3. Decomposition [see above]
4. Algorithms [see above]

However, researchers' understanding of CT does not always align with those of teachers. Cabrera (2019), in a review of literature on teachers' preconceptions of CT, notes that teachers often conflated CT with other concepts, and that an effective CT learning experience should consider what conceptions teachers enter with. Research on CT also includes work that looks at what teachers understand about technical language in CT, and what pedagogical content knowledge they possess for teaching it (Munasinghe et al., 2021; Yeni et al., 2021).

1.3. Current State of Research on Computational Thinking Teacher Education

A growing body of research has investigated effective approaches for introducing teachers (both pre-service and in-service) to CT through the design of effective coursework and professional development (PD) workshops. Several studies used the model of ongoing PD experience with sessions spanning the course of a year (Ketelhut et al., 2020; Yadav et al., 2018), leaning into the idea that effective PD should be an ongoing experience, not a single isolated session (Darling-Hammond et al., 2017). Other studies have emphasized the importance of including CT within the teacher preparation curriculum (Mouza et al., 2017; Yadav et al., 2014) so that teachers do not need additional workshops or supplementary coursework to develop a working knowledge of CT. Our work builds on these previous efforts by situating our CT module within a pre-existing mathematics & science methods course, as well as delivering it over a five-week period rather than a single workshop.

1.4. Research Questions

The present work is inspired by the potential of CT to enhance STEM education, and the need to develop additional instructional materials to introduce STEM teacher candidates to CT. Additionally, we seek to contribute to work on teacher conceptions of CT by closely examining candidates' developing conceptions of CT before, during, and after instruction. We pursue two research questions related to the design of effective CT teacher preparation, in particular how specific features of the embedded CT module shaped teachers' perceptions and understanding of CT. Two research questions are posed here because we are interested both in the initial moment of experiencing CT for the first time and what teacher candidates take away from the module as a whole, as well as how those conceptualizations change when exposed to existing academic CT frameworks.

RQ 1) How does an experience-first module on CT shape teacher candidates' emerging CT understandings?

RQ 2) After exploring this module, how do preservice teachers operationalize their own conceptualizations relative to existing CT frameworks?

2. Design

This paper presents a module of five lessons designed to introduce math and science teacher candidates to foundational ideas of CT. Here, we describe the module and underlying instructional approach before presenting results from a classroom implementation in the following section. The CT module is organized in a five-lesson sequence, occurring over a period of 5 weeks, with lessons moving from exploratory to more formal. This overall design takes inspiration from the Experience First, Formalize Later (Stats Medic, 2018) model. Experiential learning draws on the situated learning perspective, which holds that learning is a social process involving interaction with legitimate experiences in the learning subject (Lave & Wenger, 1991). Experience-based learning has been successfully used in pre-service teacher education before, in the context of cultural education (Gao, 2015). We also draw on Desimone's (2009) best practices for PD, which include participants having the chance to actively engage with the material, and ongoing, rather than isolated, sessions. This design will allow us to see how teacher candidates draw on existing knowledge to respond to discussion prompts and CT tasks. Each lesson is designed to last two hours, once a week for five weeks, meaning the entire sequence will take ten hours of class time. Participants were all fourteen teacher candidates enrolled in a mathematics and science methods course for prospective middle school (Grades 6-8) math and science teachers. This course is situated in fall of the students' Senior years, as they are simultaneously completing their part-time classroom internship experience before their full-time internship in the spring. At this point, students have taken much of their coursework and thus are relatively familiar with concepts in STEM education.

2.1. Lesson 1: Computational Thinking Activities

The first lesson is intended as an exploratory introduction to CT. Teacher candidates are asked to pair up and work through two CT activities:

(1) A Sphero robot programming task (Weintrop et al., 2022), in which participants use a mobile device connected to a rolling robot. They first use pre-existing code to roll the robot 1 meter, and then modify that code with a loop to measure the width of the classroom.

(2) The Purple Bugs CT assessment (Weintrop et al., 2014), in which participants view a simulation of bugs and grass growing in a controlled environment and explore different outcomes when different parameters are modified (for example, the speed at which grass grows or the amount of grass that the bugs eat).

These CT activities were selected because they have been used successfully to support STEM education in previous research.

For a take-home assignment, candidates are tasked with creating an "animated" Flipbook with notecards showing a ball dropping from the top of the page to the bottom. Finally, the candidates are asked to reflect on the flip book activity, as well as the two classroom activities and connections they see across the activities at this point. This will be followed up with a brainstorming task to gather candidates' initial impressions of CT in Lesson 2.

2.2. Lesson 2: Brainstorming definitions of Computational Thinking

The second lesson asks candidates to reflect on the first module's activities. The goal is to use the prior activities as a set of shared experiences that candidates can use to ground their emerging conceptualizations of what CT might be. The lesson begins with a small-group brainstorming session: each group is asked to generate a list of ideas about what CT is and what concepts and practices it might include. Following the brainstorming sessions, groups share their ideas with the class. During the whole-group discussion, the goal is to draw out and highlight the different emerging perspectives on CT shared by different candidates. The aim of this exercise is not to present a specific, concrete definition but instead to map out the landscape of potential concepts, practices, and ideas that might constitute CT.

The next activity is an introduction to Anotemos (Herbst, Chazan, & Lavu, 2019), a video-tagging tool used to facilitate analysis and discussion of classroom video. For their homework, candidates are asked to review a classroom video of 4th-grade students engaged in a CT-rich mathematics activity. Specifically, they are asked to tag moments in a video when they think the student is doing or using computational thinking as they work on a Sphero robot challenge. The results of their tagging the video will serve as materials for the Lesson 3 activity.

2.3. Lesson 3: Computational Thinking Frameworks and Relations to Standards

This lesson begins with a review of participants' tags in the assigned video. First, the tags are made publicly visible so that the participants can all see each others' tags. Review of these tags proceeds in a Think-Pair-Share format: First, participants are asked to review the class's tags and, on their own, reflect on emerging patterns and if/how their coding did or did not cohere with their classmates'. The goal of this activity is to give candidates additional experience thinking about and identifying CT in a classroom context and to further develop their conceptualizations of what CT is (or might be).

Following this discussion, the class is introduced to two formal CT frameworks: the PRADA framework (Dong et al., 2019) and the CT in Math & Science Framework (Weintrop et al., 2016). Other academic conceptualizations of CT exist; however, these two frameworks are specifically aimed at organizing concepts in CT for integration into K-12 education. Concepts in CT are grouped and phrased in ways meant to be understandable and applicable for K-12 teachers. The emphasis of this module is not that teacher candidates walk away with academically correct understandings of CT according to either of these frameworks but rather to introduce some language and structure around what constitutes CT. To emphasize this view of the frameworks, the instructor leads a discussion asking candidates to situate their emerging ideas of CT relative to the frameworks, focusing explicitly on where there are differences between the two. Part of the effort here is to empower teachers to have their own ideas of CT independent of what specific frameworks state. Related to this, the instructor highlights how there is not a specific, universally agreed-upon definition for CT and that it remains a contested term.

Following the discussion of formal frameworks, participants are asked to compare CT to standards in their content area: the Common Core State Standards (CCSS) for future math teachers and the Next-Generation Science Standards (NGSS) for future science teachers. To facilitate this process, candidates are asked to reflect on how the disciplinary standards do and do not align with CT.

This lesson concludes by introducing the activity that is the focus of the remaining two lessons: integrating CT into a classroom lesson that the candidates will teach as part of their upcoming internship. To start this culminating activity, participants group up by discipline and review 2-3 lesson plans in their subject area to brainstorm potential ways to integrate CT into each lesson plan. For homework ahead of Lesson 4, the candidates are asked to integrate CT into a single lesson and bring it with them to the following class.

2.4. Lesson 4: Lesson Plan Review

The penultimate lesson begins with a review of the CT-integrated lesson plan modifications created as the weekly assignment from Lesson 3. Participants partner up (preferably, partners will both be from the same discipline) and share the lesson plan they modified, highlighting places where they integrated elements of CT into the lesson. After the partner discussion, participants rejoin as a whole class, to share insights from their discussions and identify emerging patterns and challenges they encountered.

Following the whole-class discussion, the class examines a CT-infused lesson plan from the Sphero.Math project (Weintrop et al., 2022). The goal is to have candidates analyze an exemplary CT-infused lesson as a way to identify effective strategies for integrating CT into existing lessons and further develop candidate knowledge around ways to integrate CT into the classroom. The class concludes with another class discussion around instructional strategies for integrating CT across the curriculum. Candidates are given the assignment of revising the CT-integrated lesson plans they created for this class ahead of the final lesson.

2.5. Lesson 5: Lesson Plan Presentations and Conclusions

The final lesson of the 5-lesson sequence is focused on candidates sharing the lessons they designed that integrate CT into their content area. Each candidate is given a chance to briefly present their lesson to the class, specifically highlighting what CT concepts/practices they are focusing on and how they are integrated alongside the disciplinary content goals or the lessons. After the presentations are complete, the instructor leads a summative discussion on the relationship between CT and disciplinary content in mathematics and science classrooms.

3. Methods

The five-lesson module was piloted in a pre-service middle school (Grades 6-8) math and science methods course offered at a large university in the Mid-Atlantic region of the United States. All fourteen teacher candidates participated in our study in the Fall 2022 implementation examined here.

3.1. Data Collection

A variety of data was collected across the five classes of the methods course (Table 1). Here, we briefly describe each data source that was analyzed for this work.

Table 1
Summary of data collected throughout the implementation of the modüle

Lesson	*Video Recordings*	*Artifacts*	*Surveys*
1	Group work on CT activities		Pre-Assessment
2	Small group discussion on CT; Whole-class on CT	Brainstormed list on "What is CT?"; Anotemos tags	
3	Whole-class discussion of CT tags; Whole-class on CT frameworks; Discipline groups on Standards		
4	Partner discussions on lesson plans; Whole-class on lesson plans; Group work on Sphero lesson; Whole-class on revisions	CT Lesson Plan (1st draft)	
5	Lesson plan presentations; Whole-class concluding discussion	CT Lesson Plan (final draft)	Post-Assessment

3.1.1. Pre- and post-assessment survey

Candidates completed a questionnaire, including both Likert-scale responses and open-ended responses, that asked about their attitudes and understanding of CT, including whether they believe it can be integrated into their future classrooms (Yadav et al., 2014). The survey begins with an open-answer response question: "What is computational thinking, in your own words and to the best of your understanding?" This question was placed at the beginning of the survey to avoid participants' responses being skewed towards terms seen on the Likert items. This questionnaire served as a pre-assessment and post-assessment of these attitudes and the teacher candidates' knowledge of CT.

3.1.2. Video recordings

All five lessons of the module were videotaped, including both whole-class discussions and grouped discussions and brainstorming sessions.

3.1.3. Candidate-created artifacts

Several artifacts were collected throughout the module as indicators of candidates' understanding of CT: a brainstormed list answering "What is CT?" from Week 2, Anotemos video tags from Week 3, and the draft and final lesson plans that the candidates submitted in Weeks 4 and 5.

3.2. Analysis

Analysis of the video data and artifacts was conducted using techniques from both inductive and deductive coding approaches (Bingham & Witkowsky, 2021). We labeled passages in the data with codes generated both inductively and based on concepts and skills defined in previous academic work on CT and its domains (e.g., PRADA). For example, if a passage in a candidate's response was closely aligned with the previously defined concept of "decomposition," we used that for our code; however, we anticipated needing inductively-generated codes outside of those from previous research.

3.2.1. RQ1: Initial knowledge

To answer the first question, *How does an experience-first module on CT shape teacher candidates' emerging CT understandings?*, we concentrated on initial expressions of ideas about CT (i.e., at the beginning of the module) as these were shaped by candidates' firsthand experiences with CT activities. We focus on the first two weeks of the module, during which candidates wrote their interpretations of what CT is on the pre-assessment survey (before they have experienced the activities) and brainstormed a list answering the question "What is CT?" with their small group (afterwards in Week 2).

We began with written responses to the pre-assessment and analyzed the text with an eye for concepts and skills that candidates associated with CT. One researcher generated a list of codes to describe the data. We then reviewed this list as a research team and confirmed that the descriptions of participants' written responses appropriately capture the data, allowing members of the team with varying perspectives to share their interpretations.

The next source of data comes from Week 2, in which candidates discussed "What is CT?" first in small groups and then as a class. The analysis followed a similar process to the analysis just discussed and was conducted on transcripts from the discussions. Again, one researcher took the lead on creating a list of codes to describe the commentary from the videos; these codes again covered both elements (e.g. concepts, skills) of CT, and how those elements are organized.

The final piece of data from the module for initial impressions was from the "What is CT?" brainstormed list, representing the final product of the discussion held in each small group. We reviewed the list as written by each group.

3.2.2. RQ2: Effects of the experience first, formalize later module

To answer the second research question, *After exploring this module, how do preservice teachers operationalize their own conceptualizations relative to existing CT frameworks?*, we focused on the knowledge expressed over the latter half of the module and after its completion.

The first source of data came from Week 3, when disciplinary groups compare CT to disciplinary standards. Because we are explicitly asking candidates to organize knowledge through comparison, we anticipated that this would be a major moment of reorganization. We looked closely at these videos (one from the science group, one from the math group), attending to ways candidates expressed knowledge around CT and how they interacted with the frameworks presented. As in previous analyses, one member of the research team generated codes on a first pass, collectively reviewing with the rest of the team to confirm and modify codes.

The second source of data comes from Week 4, when candidates discussed with a partner and then the class their initial ideas for a CT-integrated lesson. Here, we paid close attention to what "domains of CT" candidates list to the discussion question, as well as how they distinguish CT from standards in their field. Again, we looked primarily for any changes in candidates' knowledge compared to the beginning of the course.

4. Results

In the results below, pseudonyms are used to protect participants' identities.

4.1. Initial Knowledge

The Week 1 pre-assessment survey question "How would you explain the concept of 'computational thinking'?" shed light on candidates' initial impressions before even experiencing the CT activities. Because candidates have had no preparation before this survey, this question measures their intuition about CT. Of the twelve analyzed candidate responses from the pre-assessment, seven distinct codes emerged (Table 2). While three gave a response coded as unfamiliar, the majority of the class gave a response indicating that they had a working concept for CT.

Table 2
Summary of codes used to label participant responses to the pre-assessment survey

Code	*Number of Participants*	*Example Text*
Algorithms/Steps	6	"Thinking methodically, almost like thinking like a calculator (step by step)." -Laura
Unfamiliar	3	"I have no idea what it means." -Ben
Calculation	2	"I think it means thinking in terms of mathematically or doing math mentally." -Jessica
Reasoning & Logical Thinking	2	"Thinking with reason and in steps. Thinking this the purpose that one thought will lead to another and continue on until you reach a conclusion." -Callie
Analysis/Data Analysis	1	"A form of logical thinking where the subject would be able to analyze and form opinions based on the information provided or collected." -Lottie
Problem Solving	1	"To solve a problem using specific steps." -Taissa
Conceptualization/Abstraction	1	"To me, this means how do our brains conceptualize how to calculate things like distance, area, or time." -Travis

Six of the participants emphasized "steps," often framing CT as aligning with the concept of an algorithm. Akilah defined CT as "*Thinking in a systematic step by step order*," which strongly resembles the idea of a step-by-step procedure denoted by "algorithm." Callie, on the other hand, described CT as "*Thinking with reason and in steps. Thinking… that one thought will lead to another and continue on until you reach a conclusion.*" While this response also includes "steps," it also incorporates the concept of a logical progression, rather than a series of steps in a task-oriented process, that is typically not included in the definition of an algorithm. This focus on reasoning was shared by another participant, Lottie, who wrote, "*I hypothesize that it would be a form of logical thinking where the subject would be able to analyze and form opinions based on the information provided or collected*." Lottie did not include any mention of step-by-step procedures, but does combine her idea of "logical thinking" with some form of analysis, of "information provided or collected" or perhaps data. Another use of "steps" comes in Simone's response, who wrote, "*The way you think is based on a process like there are steps like the scientific method*" which pulls in the idea of the scientific methods as a way of grounding her interpretation of CT. In this response, we start to see a connection to disciplinary thinking that will become a focus over the course of the lessons. These connections to different disciplinary ideas like the scientific method and logical progression to a conclusion are all connected through the same word, "steps." Though this term is closely tied to the CT concept of an algorithm, when used as a bridge to the different disciplines, it takes on a different meaning and loses the connection to algorithms.

Of the final two responses, both discussed calculation (either mental math or "how our brains conceptualize how to calculate things like distance, area, or time"). Between these and comments involving the scientific method and logical reasoning, we see that candidates' conceptions of CT are largely grounded in ideas from their respective disciplines.

Our predominant source of answering RQ1, *How does an experience-first module on CT shape teacher candidates' emerging CT understandings?*, comes from Week 2. In this lesson, candidates are again prompted to share what they think CT is by brainstorming lists of what the concept might include. As part of this activity, participants created Google Jamboards, brainstorming pages where participants can add drawings, text, images, or sticky notes, to record their thoughts before presenting to the class. These offer insight into how candidates are conceptualizing CT after their initial impression from the activities in Lesson 1. Two example Jamboard slides are presented below (Figures 1 and 2).

Figure 1
The brainstorming Jamboard created by the group of Akilah and Callie

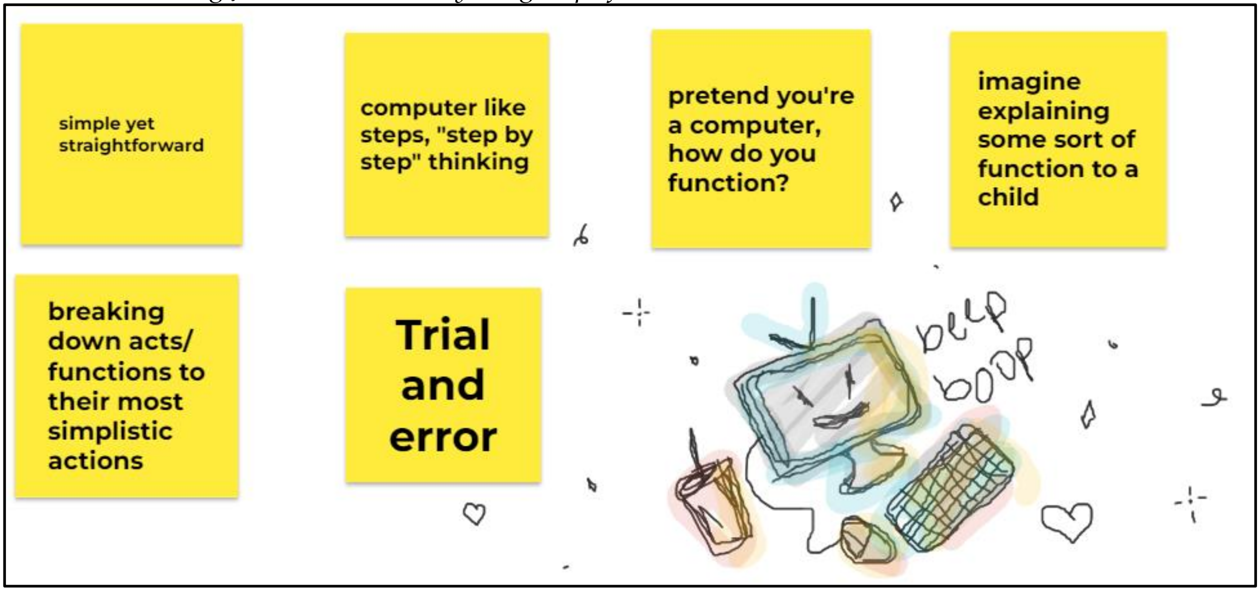


Figure 1 shows Akilah and Callie's Jamboard that articulates several ideas about CT. We labeled this slide with several codes reflecting the content of different notes: "Algorithms/Steps" for the post-it on step-by-step thinking, "Thinking Like a Computer" for the note on "*pretend you're a computer, how do you function?*" (as well as the accompanying artistic representation of a computer saying "beep boop," which Akilah drew using her tablet and a digital stylus), "Simplicity/Clarity" for the notes "*simple yet straightforward*," and "*imagine explaining some sort of function to a child*," "Decomposition" for the note on breaking down acts and functions, and "Trial and Error" for the corresponding note.

The relationship between these codes was clarified by what Akilah and Callie contributed to the class discussion; Akilah discussed including the drawing of a computer because computational thinking is:

> "very computer-like, hence the computer… If you were pretending to be a computer, like, how do you function? Like, how do you write the steps down? Kind of like thinking about how like, when you're coding, there's like a step-by-step… Like, when you're trying to make something happen, like, you write all the little steps to it. So it's like, it's simple, but it's super straightforward… we also talked about like, what if you're trying to describe something, explain some sort of function to a child, like, you would really have to, like, break it down step-by-step so that they would understand it."

Here, we see a connection between several pieces of knowledge. First, Akilah connects the notion of "computer-like thinking" concretely to two other ideas: coding, and through coding, writing well-specified algorithms with steps and procedures. The idea of specifying algorithms is also linked to the concept of simplicity, both explicitly and also using the allegory of explaining something "to a child," in which she notes it is necessary to "break things down" (which we deem close enough to the concept of Decomposition, as defined by the PRADA framework, to label as such).

These connections were also brought up by another group. Laura and Simone, in describing their brainstorm, shared:

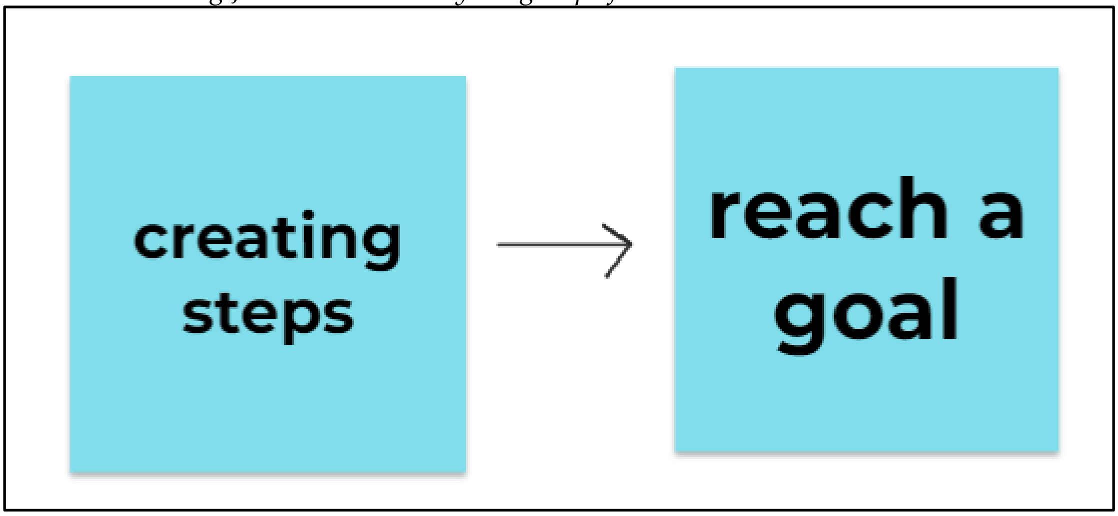
> "So for both of us, we like, really liked the term you used algorithm, because we both were thinking like, computational thinking, like computer like, and then I like, kind of brought it back to the Sphero robot things where it was like, it almost felt like coding on Scratch. And like, I always think of coding as very, like, step-by-step thinking. And that also relates me back to, like, an algorithm."

Again, the idea of an algorithm is linked to coding. Further, when responding to press from the instructor on what "computer-like" meant, Laura also talked about how coding a computer involves very small, specific functions:

> "They do specific things if you tell them to do it. You know, like you code a computer in a way-when you use like, a keyboard, like backspace when you type, or like, when you're typing something, you're telling it to show the image of the letters on the screen. So I think like, it's very much like you tell someone to do something in directions, and then they do it back. So it's like very, it's step-by-step algorithm. And like, it'll give you that result with that input."

Here, Laura again mentions the idea of an algorithm and connects it to coding, and through coding to an allusion of specific functions governing small actions within a computer's overall programming.

Figure 2
The brainstorming Jamboard created by the group of Ben and Lottie



Ben and Lottie explicitly showed a connection between two concepts in their Jamboard slide (Figure 2), connecting "creating steps" with "reach a goal" via an arrow. We interpreted this link to mean that creating steps is done in the service, or for the purpose of, achieving some goal. Ben's commentary supports this interpretation as he described how in each activity, we created steps in service of a larger goal:

> "For the ball rolling around like you had to create, like, the, the coding steps or whatever. And then our goal was to reach... The goal we're reaching was to see how wide the room was or whatever. And then for the ... Oh, yeah, so the bugs, it was like, we were creating steps to see like, if we change certain things like, what happened to the boat, like we were changing, like, how much water the plants got, or whatever, like changing the drought or whatever. And then we were trying to see how it affects the end goal and things like that. And then for the flipbook, it was kind of the same way, we're creating steps like reading the flip of the pages to see what it would look like."

In each activity, Ben sees some form of "creating steps" that is in service of some larger end. With the Sphero robot, he saw this as coding a procedure for the robot to follow to measure the room. For the Purple Bugs activity, he saw this as changing the parameters of the simulation to get a different result. For the flipbook, he saw the creation of each page as a step in service of the overall animation. In each case, there are smaller steps (reminiscent of Decomposition), but also a focus on an end goal or product. In Ben's account, CT is creating (or perhaps modifying) steps to *produce* something, whether a measurement of a room, a simulation of an ecosystem, or an animated flipbook.

4.2. Changing Knowledge

To answer RQ2, *After exploring this module, how do preservice teachers operationalize their own conceptualizations relative to existing CT frameworks?,* we looked at data from later in the module. In the third lesson of the module, candidates are presented with two CT frameworks: the PRADA framework (Dong et al., 2019) and the CT in Math and Science Taxonomy (Weintrop et al., 2016). Participants took quickly to the new academic frameworks, noting mostly similarities between the frameworks and ideas they had already discussed about CT. For example, Jessica described CT as an "umbrella" described by their previous ideas that also included the domains and practices in the frameworks:

> "The stuff that we already identified in the Jamboard are things that are mentioned here: observing, trial and error… all this stuff is under the umbrella of CT. So it's not like we're ignoring- we know what CT is, we're just putting more concrete, like, definitions to add to our understanding."

Jessica had previously focused heavily on trial and error as a key element of CT. In Week 2, she described CT as "*using our experiences and a trial and error kind of process*," listing examples from each of the Week 1 CT activities:

> "And then as we were like, in the class we had tested, which works and what doesn't and using those experiences to be like: Okay, well, how far do you think it has to go in order to measure like, the distance of the block? Or like with the Purple Bug thingy? Like, adjusting the amount of drought or how much grass decreased during the drought and how that affects like, the bug population making predictions based on like, oh, like, if there are less grass [tiles] that means more bugs will die because it- there's no food, etc. And then also like the trial, full trial and error process with kind of like, an animated flipbook, where we're kind of like, okay, well if it does- didn't work this way, drawing that way doesn't work, then like, how can we make our instructions more specific?"

Jessica sees this trial and error process in both frameworks. She sees it as fitting with "Algorithms" and "Decomposition" on the PRADA side, while fitting into "Systems Thinking" on the Taxonomy side. For her, the introduction of the frameworks represents a shift in terminology. She does not see the frameworks as offering new ideas about CT; rather, they merely offer new ways to talk about existing ideas.

Laura, in contrast, describes changes in her thinking about CT prompted directly by the frameworks:

> "When I was doing the Jamboard, thinking about what computational thinking was after we did, like, the Sphero and everything, I very much focused on the algorithms part. I really thought, like, that is computational thinking. And now that I'm seeing the rest of PRADA, like all the other things, I can see how there's things that we were doing that would fit into, maybe, this definition of computational thinking."

In contrast to Jessica's relatively minor, vocabulary-centric reconceptualization of CT, Laura's realization seems to be a deeper adjustment. Rather than seeing CT as defined solely by algorithms, she has reorganized her understanding so that algorithms are only one of several ideas under the CT umbrella. This change, then, not only adjusts the pieces of knowledge she associates with CT (based on her mention of PRADA, we assume that she is now including Pattern Recognition, Decomposition, and Abstraction) but also the reorganization of a previous piece of knowledge as now just one domain among four.

As part of the summative activity for the CT module, 12 lesson plans were submitted and analyzed. The topics of these lessons, and the CT domains or practices integrated into each, are summarized in Table 3.

Table 3
CT domains/practices (from PRADA and the CT in M/S Taxonomy) integrated into lesson plans

Author	*Lesson Topic*	*CT Domains/Practices*
Mari	Ratios & Proportions	Abstraction, Decomposition
Travis	Area of Parallelograms	Decomposition
Shauna	Climate Change and Natural Hazards	Pattern Recognition, Analyzing and Visualizing Data, Modeling & Simulation Practices
Lottie	Modeling Linear Relationships	Pattern Recognition
Natalie	Air Fronts & Weather	Pattern Recognition, Understanding the Relationships in a System, Defining Systems and Managing Complexity, Analyzing and Visualizing Data
Taissa	Carbon Footprint	Pattern Recognition, Abstraction, Decomposition, Algorithms
Simone	Areas of Parallelograms	Pattern Recognition, Abstraction, Algorithms
Jeff	Asexual and Sexual Reproduction	Computational Model, Abstraction, Collecting Data, Analyzing Data
Misty	Charts and Graphs to Express Relationships Between Data	Data Practices, Pattern Recognition, Algorithms
Callie	Displaying a Data Distribution	Collecting Data
Akilah	Scale Drawings of Gardens and Area	Decomposition, Modeling and Simulation, Assessing Different Approaches/Solutions
Laura	Linear Relationships: Car Speed and Skid Marks	Assessing Different Approaches/Solutions, Analyzing and Visualizing Data, Troubleshooting/Debugging, Thinking in Levels, Defining Systems & Complexity, Using & Designing a Computational Model

Candidates' lesson plans integrated a wide variety of CT domains and practices, drawn from both PRADA and the CT in Math and Science Taxonomy. Across all lesson plans, we see four domains from PRADA and eleven practices from the Taxonomy. Compared to the seven codes we found in candidates' initial commentary on CT, this demonstrates that they are including many more ideas in their conception of CT. Most candidates' lesson plans also included multiple domains or practices, indicating a shift away from a view of CT as a singular concept, as seen in many responses to the initial survey.

5. Discussion

This multi-lesson module offers teacher candidates exposure to CT through many different lenses: first through the hands-on experience of completing a series of CT activities, second through their own interpretations of those activities, third through academic frameworks, and finally through the act of creating a lesson plan that includes CT practices and ideas. Each of these also provides an opportunity for researchers to gauge participant understanding as they progress through the module.

As shown above, candidates brought prior knowledge and experiences to the module that served as a means for grounding their emerging understanding of CT. This largely focused on algorithms and "step-by-step" thinking, with some mention of calculations and the scientific method. These latter points are interesting as it shows candidates' proclivity for trying to situate

CT within disciplinary contexts. By the end of the module, candidates were using a wide array of ideas that show up across CT frameworks like PRADA and the CT in Math and Science Taxonomy. This suggests the productivity of including opportunities for learner-generated definitions (in this case, through hands-on activities and discussion) before exposure to formalized academic definitions.

5.1. Experience First, Formalize Later and Computational Thinking

A central design decision for this work is the implementation of the Experience First, Formalize Later approach (Stats Medic, 2018) to introduce teacher candidates to CT. This is a novel approach in CT teacher education, where other efforts have been structured as guest lectures (Yadav et al., 2014) or PD workshops (Ketelhut et al., 2020). The decision to use the experience-first structure was motivated by prior research that found math and science teacher candidates, when being introduced to CT through frameworks, would rely heavily on specific language and ideas that aligned with their disciplinary knowledge (Walton & Walkoe, 2021). For example, after being introduced to the PRADA framework, the authors documented how a math teacher clung tightly to Pattern Recognition as a way to interpret CT in a mathematics context given the overlaps the teacher saw. The result was a rather narrow conceptualization of CT. In contrast, with the Experience First, Formalize Later approach presented above, candidates are given hands-on CT experiences and are provided the opportunity to create their own ideas about CT based on these experiences. This allows candidates to develop their own conceptualization of CT before being introduced to formalized definitions of CT. This approach is particularly generative given the existence of several CT frameworks and ways they cohere and differ as it relates to specific concepts and practices (Shute et al., 2017).

5.2. Limitations and Future Work

The goal of this work was to document the experience and outcomes of teacher candidates being introduced to CT via an Experience First, Formalize Later module integrated into a pre-service teacher methods course. While we think the module was successful and the accompanying study was able to document productive CT reasoning from the candidates, this study is not without its limitations that we hope to address with future work. First, this sample is relatively small and all of the candidates are in the same program, meaning they all have shared educational experiences through the program they are enrolled in and live in the same geographical region. While this does not undermine what is presented above, it points the way toward future work to explore how candidates from different regions and different sets of prior knowledge would engage with the presented module. Another limitation comes from the specific activities and technologies used in the CT module. Does Sphero foreground specific CT practices? Does the Purple Bugs activity make certain CT concepts more salient? Likewise, we chose two CT frameworks to focus on but there are others that could have been used (e.g. Brennan & Resnick's (2012) framework of CT as concepts, practices, and perspectives). All of these decisions directly influence the results of the paper and it is not clear how findings might have differed with a different set of activities or frameworks. Exploring this relationship is a second possible avenue of future work. Finally, our positionality as CT researchers and math educators shaped the lens we brought to this work. Further, the fact that we designed the module, led the class sessions where candidates engaged in the module, and conducted the analysis presents a situation where we are not objective outsiders conducting the analysis but instead are deeply invested in the results. While we do not think this undermines the findings as the data presented above speak for itself, this does suggest this is a context where recruiting others to extend this work and use the module in other university environments would be particularly generative.

6. Conclusion

This work sought to explore teacher candidates' emerging conceptualizations of CT through the use of a 5-lesson module embedded within an existing elementary math and science pre-service methods course. Through engaging with a series of CT activities and discussions, candidates' understanding of what CT is (or could be) expanded. In structuring the lessons in a way that first provided hands-on experience with CT-infused activities before moving to formalized definitions, the sequencing sought to help candidates ground their emerging understanding of CT in ways that helped them build connections with prior knowledge and experience. In particular, activities that allowed candidates to engage with CT and then collaboratively reflect on those experiences and identify commonalities across these experiences served as a useful context to develop foundational intuitions related to CT before introducing them to formalized definitions and terminology. In doing so, this approach helped candidates develop flexible CT definitions that can help them infuse CT across their classrooms in a diversity of ways, as seen in the breadth of CT and disciplinary concepts included across the final projects (Table 3). Collectively, this work contributes to the growing literature on ways to introduce future teachers to CT and showcases candidates' abilities to identify CT, its potential connections to disciplinary content, and connect their prior knowledge and hands-on CT experiences to formalized definitions that can serve as a foundation for infusing CT into their future teaching.

Acknowledgements: The design of this project were presented to the 2022 Maryland Center for Computing Education (MCCE) Higher Education Summit.

Author contributions: All authors are agreed with the results and conclusions.

Declaration of interest: None of the authors have a conflict of interest in this project.

Ethics declaration: Authors declared that ethics approval for this study was obtained from University of Maryland College Park (Protocol No: 1712220) on 11 July, 2022.

Funding: This project is funded by the Maryland Center for Computing Education.

References

Abelson, H., & diSessa, A. A. (1986). *Turtle geometry: The computer as a medium for exploring mathematics*. The MIT Press.

Bih, J. S., Weintrop, D., Walton, M., Elby, A., & Walkoe, J. (2020). Mutually supportive mathematics and computational thinking in a fourth-grade classroom. In Gresalfi, M. and Horn, I. S. (Eds.), *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020,* (Vol. 3, pp. 1389-1396). International Society of the Learning Sciences.

Bingham, A. J., & Witkowsky, P. (2021). Deductive and inductive approaches to qualitative data analysis. In C. Vanover, P. Mihas, J. Saldaña, (Eds.), *Analyzing and interpreting qualitative data: After the interview* (pp. 133-146). SAGE.

Brennan, K., & Resnick, M. (2012, April). *Using artifact-based interviews to study the development of computational thinking in interactive media design* [Paper presentation]. American Educational Research Association meeting 2012, Vancouver, BC, Canada.

Cabrera, L. (2019). Teacher preconceptions of computational thinking: a systematic literature review. *Journal of Technology and Teacher Education, 27*(3), 305–333.

Darling-Hammond, L., Hyler, M. E., Gardner, M., & Others. (2017). *Effective teacher professional development*. Learning Policy Institute. http://www.oregonrti.org/s/NO_LIF1.PDF

Desimone, L. M. (2009). Improving impact studies of teachers' professional development: Toward better conceptualizations and measures. *Educational Researcher, 38*(3), 181–199. https://doi.org/10.3102/0013189X08331140

Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). PRADA: A practical model for integrating computational thinking in K-12 education. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 906–912. https://doi.org/10.1145/3287324.3287431

Gao, X. (2015). Promoting experiential learning in pre-service teacher education. *Journal of Education for Teaching*, *41*(4), 435-438, DOI: 10.1080/02607476.2015.1080424

Herbst, P., Chazan, D. & Lavu, S. (2019). *Anotemos. Web-based collaborative software tool for the annotation of video*. Disclosed to the Office of Technology Transfer, University of Michigan.

Kaput, J., Noss, R., & Hoyles, C. (2002). Developing new notations for a learnable mathematics in the computational era. In L. D. English (Ed.), *Handbook of International Research in Mathematics Education*, (pp. 51–75). Routledge.

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology*, *29*(1), 174–188. https://doi.org/10.1007/s10956-019-09798-4

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press.

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). https://ajet.org.au/index.php/AJET/article/view/3521

Munasinghe, B., Bell, T., & Robins, A. (2021, February 2). Teachers' understanding of technical terms in a computational thinking curriculum. *Australasian Computing Education Conference,* 106-114. https://doi.org/10.1145/3441636.3442311

Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings: Learning cultures and computers*. Kluwer.

Papert, S. (1972). Teaching children to be mathematicians versus teaching about mathematics. *International Journal of Mathematical Education in Science and Technology*, *3*(3), 249-262. https://doi.org/10.1080/0020739700030306

Papert, S. A. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic books.

Roschelle, J., Kaput, J., & Stroup, W. (2000). SIMCALC: Accelerating student engagement with the mathematics of change. In J. Roschelle, J. J. Kaput, W. Stroup (Eds.), *Innovations in Science and Mathematics Education* (pp. 47–75). Routledge.

Schanzer, E., Fisler, K., Krishnamurthi, S., & Felleisen, M. (2015). Transferring skills at solving word problems from computing to algebra through Bootstrap. In *Proceedings of the 46th ACM Technical symposium on computer science education* (pp. 616-621). Association for Computing Machinery, United States. https://doi.org/10.1145/2676723.2677238

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Stats Medic. (2018). Experience first, formalize later. *Stats Medic.* https://www.statsmedic.com/post/experience-first-formalize-later

Walton, M. & Walkoe, J. (2021). Computational Thinking as a Context for Ambitious Math Instruction. In de Vries, E., Hod, Y., & Ahn, J. (Eds.), *Proceedings of the 15th International Conference of the Learning Sciences - ICLS 2021* (pp. 609-612). International Society of the Learning Sciences.

Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., & Wilensky, U. (2014). Interactive assessment tools for computational thinking in high school STEM classrooms. In *Intelligent Technologies for Interactive Entertainment: 6th International Conference, INTETAIN 2014, Chicago, IL, USA, July 9-11, 2014. Proceedings 6* (pp. 22-25). Springer International Publishing.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). 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

Weintrop, D., Walkoe, J., Walton, M., Bih, J., Moon, P., Elby, A., Bennett, B. & Kantzer, M. (2022). Sphero.Math: A computational thinking-enhanced fourth grade mathematics curriculum. In *Computational Thinking in PreK-5: Empirical Evidence for Integration and Future Directions* (pp. 39-46). https://doi.org/10.1145/3507951.3519286

Wilensky, U. (1995). Paradox, programming, and learning probability: A case study in a connected mathematics framework. *The Journal of Mathematical Behavior,* *14*(2), 253–280. https://doi.org/10.1016/0732-3123(95)90010-1

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

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, *28*(4), 371–400. https://doi.org/10.1080/08993408.2018.1560550

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education, 14*(1), 1-16. https://doi.org/10.1145/2576872

Yeni, S., Grgurina, N., Hermans, F., Tolboom, J., & Barendsen, E. (2021, October). Exploring teachers' PCK for computational thinking in context. In *The 16th Workshop in Primary and Secondary Computing Education*, (pp. 1-10). https://doi.org/10.1145/3481312.3481320