Barlow, E. K., Barlow, A. T., & Nadelson, L. S. (2023). Computational thinking: Perspectives of preservice K-8 mathematics teachers. *Contemporary Issues in Technology and Teacher Education, 23*(2), 337-367.

Computational Thinking: Perspectives of Preservice K-8 Mathematics Teachers

Elizabeth K. Barlow
Auburn University

Angela T. Barlow & Louis S. Nadelson
University of Central Arkansas

Advancements in computing have led to increased interest in integrating computational thinking in the K-12 curriculum. Computational thinking can be defined as a problem-solving process with the goal of developing algorithms that can be coded for computer use. With its emphasis on problem solving, the processes associated with computational thinking overlap with those of mathematical thinking, leading to an anticipated reliance on mathematics teachers to teach computational thinking in the K-12 setting. Currently, research related to preservice mathematics teachers' perceptions of computational thinking is emergent; yet, this research is needed to inform leaders of teacher preparation programs. The purpose of this study was to investigate preservice K-8 mathematics teachers' views of teaching computational thinking. Participants from three different universities completed an asynchronous, online simulation, responding verbally to prompts related to the importance of and processes for teaching computational thinking to all students. Results demonstrated that participants found value in teaching computational thinking, although the majority either did not connect their ideas specifically to computational thinking or erroneously connected their ideas to mathematical computations and/or technology integration. Further, a large majority of participants demonstrated deficit perspectives of students considered lower achieving. Implications and areas for future work are included.

Given rapid advancements in computing, educating today's students necessitates preparing them to utilize technology for addressing real-world problems (Association of Mathematics Teacher Educators [AMTE], 2017; National Council of Teachers of Mathematics [NCTM], 2014; Sykora, 2021). Thus, educators must develop students' skills in computational thinking, which is a problem-solving process that involves formulating and solving complex problems with the goal of automating methods for use in solving related problems (International Society for Technology in Education [ISTE] & Computer Science Teachers Association [CSTA], 2011).

Computational thinking is considered a cross-curricular skill, as it may be used to solve problems associated with diverse disciplines (ISTE, 2021). Given the requirements of computational thinking to engage in problem posing, problem solving, and mathematical modeling (ISTE & CSTA, 2011), mathematics courses are likely to be considered an appropriate context for advancing student knowledge of computational thinking (Gadanidis et al., 2017; Kallia et al., 2021; Rycraft-Smith & Connolly, 2019; Weintrop et al., 2016).

As computational thinking continues to gain momentum in K-12 education, the overlap with mathematical processes suggests K-12 mathematics teachers are likely to have a role in supporting student attainment of this cross-curricular skillset (Weintrop et al., 2016). As part of this work, mathematics teachers will need to advance equity as they "proactively counter stereotypes that exclude students from opportunities to excel in computing" (ISTE, 2021, "Equity Leader" section). As described by the National Association of the Education of Young Children (NAEYC, 2019), advancing equity includes a commitment to equitable learning opportunities for all children, a position that must be applied to the teaching and learning of computational thinking. As a result, mathematics teacher preparation programs must consider their role in preparing future mathematics teachers to effectively teach computational thinking to all students.

With this as our backdrop, the purpose of this study was to explore preservice K-8 mathematics teachers' views on teaching computational thinking. Specifically, the research question was, How do preservice K-8 mathematics teachers view the importance of teaching computational thinking, and what is the role of equity in their views? Examining preservice teachers' views provides a foundation on which to develop curriculum, activities, and experiences to advance their knowledge of teaching computational thinking to all students. In addition, when contrasted with mathematical thinking, these insights serve to inform the work of mathematics teacher educators.

Review of Literature

There is a growing number of initiatives to integrate teaching computational thinking across the K-12 curriculum (Kafai & Proctor, 2021). Although the initiatives may be motivated by a desire to increase the number of people prepared to pursue careers in computing, there are multiple challenges associated with teaching computational thinking. One of the greatest challenges lies within teacher preparation (Cuny, 2011).

The lack of or limited curriculum in teacher preparation programs for teaching computational thinking (Mason & Rich, 2019) suggests that many teachers have not had formal preparation to teach the concepts and processes. However, some parts of the K-12 curriculum, such as mathematics, appear to be aligned with computational thinking. As a result, there is a perception that mathematics teachers are prepared to teach computational thinking (cf. Gadanidis et al., 2017) despite the lack of formal preparation to do so.

Although teachers of mathematics may not have received formal preparation to teach computational thinking, their preparation likely includes concepts and processes that overlap with teaching computational thinking (Hsu et al., 2018). The overlap may leave the impression that their mathematics teacher preparation coursework provides a foundation for teaching computational thinking. Therefore, it is likely that there will be high levels of reliance on mathematics teachers for teaching or integrating computational thinking as part of mathematics instruction (Weintrop et al., 2016). This reliance, in turn, leads to the need to define and compare computational thinking and mathematical thinking.

Computational Thinking

There are multiple views regarding what constitutes computational thinking, many of which rely on the seminal work of Wing (2006). The CSTA has taken efforts to synthesize the array of perspectives into a set of computational thinking standards and expectations applicable to K-12 education. Given our emphasis on preservice K-8 teachers, we embraced the ISTE and CSTA (2011) standards to convey our perspective of computational thinking.

The ISTE and CSTA (2011) definition of computational thinking involves engaging in multiple activities such as defining problems in ways that align with the logic of computers, organizing data logically, creating and applying models and simulations of data, thinking algorithmically to create programmable solutions, and applying coded solutions to related problems. It is apparent from this definition that the goal of computational thinking is for students to examine problems logically, systematically, and algorithmically and to apply the knowledge they gain through problem examination to create computer programs that they can use to generate solutions using different inputs.

There is a view among teachers and others that the logic and processes of computational thinking heavily align with mathematics due to the shared process of problem solving (Denning, 2005; Edwards & Cassidy, 2021; Sneider et al., 2014). However, the goals of mathematics are different and, therefore, involve processes that are distinct from the processes associated with computational thinking (Shute et al., 2017). Thus, there is a need to define mathematical thinking.

Mathematical Thinking

Although thinking about mathematics is dependent on context and culture (Tall, 1991), "conceptions of mathematical thinking appear to coalesce around a number of central principles" (Monteleone et al., 2018, p. 560). Most pertinent to our work is the definition of mathematical thinking within the K-12 mathematics education community. Therefore, we relied on the *Common Core State Standards for Mathematics* (National Governors Association Center for Best Practices [NGA Center] & Council of Chief State School Officers [CCSSO], 2010), specifically the Standards for Mathematical Practice, to communicate mathematical thinking in K-12 teaching and learning.

The Standards for Mathematical Practice (NGA Center & CCSSO, 2010) describe the processes of thinking that students (and potentially professional mathematicians) engage in when solving mathematics problems. These processes include making sense of problems through analysis and identification of relationships among variables, identifying constraints, abstracting relationships into symbols to form equations, applying equations in unique ways, and rationalizing solutions. The standards also include forming arguments and critiquing the work of others, which frequently requires finding similar relationships in unfamiliar or new situations. Engagement in these processes (i.e., mathematical thinking) catalyzes students' attainment of the goal of mathematical proficiency (National Research Council, 2001).

Mathematical Thinking vs. Computational Thinking

Both computational thinking and mathematical thinking involve problem solving, using logic, systematically identifying constants and variables, determining their relationships, creating expressions or equations that can be applied to solve related questions, and working to provide effective optimal solutions for additional applications (cf., NGA Center & CCSSO, 2010; ISTE & CSTA, 2011). Despite these commonalities, mathematical thinking and computational thinking diverge for two reasons. First, the goals associated with each type of thinking are different. The goal of mathematical thinking is to solve problems encountered in real life, mathematics, and other disciplines by identifying relationships, creating abstractions, or forming equations (NCTM, 2014). Once a problem's solution (or solutions) is achieved, emphasis is given to judging the reasonableness of the solution or justifying the mathematical ideas through formal proof.

Like mathematical thinking, computational thinking involves processes that lead to a problem's solution or solutions. Once a solution to the problem is found, though, the goal of computational thinking is to reflect on the solution process and translate it into an algorithm that is aligned with computer logic (Shute et al., 2017). Algorithms can be coded into programs and run on computers to solve related problems using different inputs. Although some mathematical solutions can be translated into computer programs, other mathematical solutions cannot be coded due to the possibility of multiple outcomes that cannot be defined using algorithms.

For example, a common task for middle school mathematics students is to determine under what condition three segments with given lengths can be used to create a triangle. With this task, students explore and typically conjecture that the sum of the side lengths of any pair of sides must be longer than the length of the remaining side. With this conjecture in hand, mathematics students set out to prove the resulting theorem referred to as the triangle inequality theorem. In contrast, if students were to engage in computational thinking with this task, with conjecture in hand, students would attempt to write a code so that the computer could determine whether a triangle was possible based on the side lengths as inputs into the code.

The second reason that mathematical thinking and computational thinking diverge is related to the concepts and practices to be learned in each area. In an exploration of standards, Rich et al. (2020) noted processes that on the surface seemed to be common to both mathematical thinking and computational thinking; however, as they worked to make sense of these processes, important differences were noted.

An example offered by Rich et al. (2019) involved completeness. When considering completeness in mathematics, students are expected to justify their thinking and explain their understanding. In contrast, in computational thinking completeness refers to the inclusion of all steps in a solution process so that a code can be created that a computer can interpret and execute. Similarly, other authors have noted surface level commonalities between computational thinking and mathematical thinking that represent significant differences that, if not addressed, serve as barriers for students' understanding of key mathematical or computational concepts (e.g., Bråting & Kilhamn, 2021).

Given these differences in goals and concepts, the perception of significant overlap in the two ways of thinking has the potential to negatively influence students' learning in each area. Thus, there is benefit in acknowledging the ways of thinking as distinct. Our position differs from others' efforts to delineate the similarities and differences between computational and mathematical thinking. For example, in describing the similar aspects, Shute et al. (2017) included problem solving, modeling, data analysis, and statistics. When describing the unique aspects of mathematical thinking, though, Shute et al. listed the disciplines within mathematics, such as algebra and geometry. Their model did not take into consideration the previously described differences between mathematical and computational thinking. Further, their model did not differentiate between the processes of mathematical thinking and the disciplines within mathematics (cf. NGA Center and CCSSO, 2010).

Similarly, well-respected computer scientists have failed to draw a distinction between computational thinking and mathematical thinking (e.g., Turner & Angius, 2017), which reflects a lack of understanding of mathematical thinking. Clearly delineating these similarities and differences is fundamental to preparing teachers to teach both mathematical thinking and computational thinking, as the assumption that mathematical and computational thinking are nearly identical is likely to lead to barriers and increased challenges in preparing teachers.

Preparing Teachers to Teach Computational Thinking

As with other content areas, teaching computational thinking requires preparation (NRC, 2010). Without proper preparation, there is a potential for teachers to teach computational thinking based on their personal perspectives, preconceptions, and misconceptions (Chang & Petterson, 2018; Edwards & Cassidy, 2021). For example, common misconceptions of computational thinking include conflation with numerical computations (Fessakis & Prantsoudi, 2019), particularly if computational thinking is taught in the context of mathematics (Bouck et al., 2021; Edwards & Cassidy, 2021).

In addition, teachers may equate computational thinking with technology integration (Cabrera, 2019; Edwards & Cassidy, 2021) and miss opportunities to engage students in the processes associated with computational thinking. Thus, integrating computational thinking content into preservice teacher preparation programs seems to be critical to ensuring teachers have accurate knowledge of computational thinking.

Bower and Falkner (2015) surveyed 44 pre-service teachers, with 38 indicating they were going to teach elementary school and five planning to teach at the secondary level. The authors reported that the preservice teachers in their sample possessed a weak understanding of computational thinking, as the participants provided conflated or broad definitions that demonstrated great potential to improve their understanding of computational thinking. This result was limited, though, due to the smaller sample size and the context of a single university in Australia.

Other research has demonstrated that relatively brief interventions may be useful for raising teachers’ awareness of ways to conceptualize computational thinking. For example, Gadanidis et al. (2017) shared findings from their analysis of preservice teachers’ assignments and reflections from a 9-week course on computational thinking in mathematics education, in which they found that the intervention allowed the preservice teachers to more accurately conceptualize how computational thinking fits in the K-12 classroom. However, these interventions were not sufficient for preparing teachers with the deeper understanding needed to effectively teach computational thinking (see also Mouza et al., 2017; Yadav et al., 2014). Thus, to prepare preservice teachers to teach computational thinking effectively, there is a need for content that is strategically designed to address misconceptions, with concentrated focus on computational thinking and long-term engagement in its processes (Fessakis & Prantsoudi, 2019; Yadav, Gretter et al., 2017; Yadav, Stephenson et al., 2017).

Equity in Teaching Computational Thinking

Shulman and Shulman (2004) identified five factors that influence teacher development; among them was vision. In this context, a preservice teacher’s vision represents their view of an ideal classroom, which is influenced by their experiences, values, and assumptions (Hammerness, 2003). To heed the calls of ISTE (2021) and NAEYC (2019), preservice teachers' visions of an ideal classroom must include elements that advance equity.

We (the authors) acknowledge that to achieve equity, the institutional and systemic structures that lead to inequities must be dismantled (Diversity in Mathematics Education Center for Learning and Teaching, 2007; Martin et al., 2017). The ongoing presence of these inequities, though, signals the importance of considering the teacher's role in advancing equity, as teachers' beliefs about students and their curriculum decisions represent a means for advancing equity (NCTM, 2014). To advance equity is to be committed to equitable learning opportunities for all students, regardless of student attributes (e.g., gender, race/ethnicity, prior achievement; NAEYC, 2019). By equitable learning opportunities, we mean providing access to challenging content along with the supports needed to be successful (NCTM, 2014).

To advance equity in this regard, preservice teachers must possess not only a belief that all students are capable of learning challenging content but also a commitment to differentiating instruction to support students in this quest (AMTE, 2017). Unfortunately, it is well documented that preservice teachers often hold deficit views of students (e.g., Sleeter, 2008), particularly in mathematics where low achievement is often explained through deficit models (NCTM, 2014).

Deficit views of students in mathematics classrooms can lead to modifying instruction to make it less challenging (Battey & Stark, 2009; Milner, 2012) or developing different expectations with regard to learning outcomes (Milner, 2012; Polard, 2013). As a result, deficit views serve as obstacles toward advancing equity in mathematics. One must wonder, then, whether preservice teachers hold views that will advance or hinder equity in terms of computational thinking.

Summary and Current Study Significance

Preparing teachers to teach computational thinking effectively involves developing a deep understanding of computational thinking (Fessakis & Prantsoudi, 2019; Yadav, Gretter et al., 2017; Yadav, Stephenson et al., 2017), with attention given to how it differs from mathematical thinking. Unfortunately, there is a dearth of teacher preparation curriculum related to computational thinking, as research in this area is emergent (Mason & Rich, 2019). Further, despite the aforementioned differences in goals, concepts, and practices (e.g., Rich et al., 2020), not all available literature recognizes these distinct differences between computational thinking and mathematical thinking (e.g., Shute et al., 2017). Combined, these ideas expose the issue of relying on teachers to teach computational thinking based on their preparation for teaching mathematics.

Without proper preparation, teachers will likely teach computational thinking based on their personal perspectives (Chang & Petterson, 2018; Edwards & Cassidy, 2021). Gathering data associated with preservice teachers' perceptions of the importance of teaching computational thinking would likely expose the barriers that should be addressed in teacher preparation programs.

Empirically documenting the barriers is particularly important when supporting new teachers who are expected to teach content that they did not learn about in their teacher preparation programs. In addition, the critical importance of advancing equity in preservice teachers' vision for teaching computational thinking provides justification for documenting their perspectives of teaching computational thinking. Collectively, these data can be used to establish context and provide direction for examining teacher instructional preparation to teach computational thinking, as well as to inform initiatives such as teaching computational thinking to all.

Methodology

Our interest in preservice teachers' views of teaching computational thinking was motivated by a desire "to get at the inner experience of participants . . . and to discover rather than test variables" (Corbin & Strauss, 2008, p. 12). As a result, the purpose of this qualitative study was to explore preservice K-8 mathematics teachers' views on teaching computational thinking. Participants responded to open-ended questions featured in an online, asynchronous simulation. We analyzed these responses using open coding (Corbin & Strauss, 2008; Creswell, 2013). Details of the methods are provided in the following sections.

Participants

Participants for this study were K-8 preservice teachers from three universities (see Table 1). We were not affiliated with these universities and did not serve as course instructors for the participants. Also, none of the universities included computational thinking in their teacher preparation programs; therefore, one could assume that the participants had little to no background in computational thinking.

Table 1
Overview of Participants

Category	University		
	A	B	C
Major			
Early Childhood	0	4	0
Elementary	7	31	26
Middle Grades	0	2	3
Special Education	0	2	3
Not specified/Other	0	1	1
Classification			
Freshmen	0	10	1
Sophomore	5	20	19
Junior	2	8	6
Senior	0	2	6
Not specified	0	0	1

The first university, University A, was a private, faith-based institution with an enrollment of approximately 8,000 students and located in an urban city in a southeastern region of the United States. Preservice teachers enrolled in a single section of a mathematics content course for teachers were invited to participate in the study. Of the 13 students enrolled in the course, seven students participated and received extra credit on an assignment.

The second university, University B, was one of three campuses associated with a large, public university in a southeastern region of the United States. Located in a coastal city, University B enrolled approximately 6,000 students. Preservice teachers in an integrated science course were invited to participate in the study. Of the 96 preservice teachers enrolled, 40 fully participated and received extra credit that was applied to their final course grade.

The third university, University C, was a public research university located in the mid-Atlantic region of the United States. Situated within a college-oriented city, University C enrolled approximately 21,000 students. For the study, 33 preservice teachers enrolled in a mathematics content course for teachers were invited to participate in the study. The instructor used the simulation as a class assignment: all 33 preservice teachers consented to have their responses included in the study for analysis.

Procedures

To capture participants' perspectives on computational thinking, we utilized an online platform called Teacher Moments (https://teachermoments.mit.edu/). This platform was designed as an instructional tool to support preservice teacher learning through engagement with scenarios in a low-stakes environment (Teaching Systems Lab, 2020). Teacher Moments provides users with the ability to capture individuals' ideas through voice recordings rather than through typed entry.

Although our purpose for using Teacher Moments was not within its original intended purpose, we believed the platform held the potential for gathering the data necessary for us to answer our research question. We thought voice recordings would enable us to gather authentic participant responses due to the ability to capture their free flow of thought. This is in contrast to the high potential for brief, edited statements that could have been received if the participants had typed their responses. The Teacher Moments interface allows participants to rerecord their responses, if desired, and immediately transcribes audio responses to text.

Within the Teacher Moments platform, we developed a scenario that placed the participant in the role of a new teacher in an elementary school setting, in which the principal announced a new initiative to focus on teaching computational thinking to *all* students. Further, the school also had a notable percentage of lower achieving students. The simulation consisted of a series of screens, each with information for the participant to read and react to using the platform's audio-recording feature.

Table 2 contains the scenario information in order of screen succession. It is worth noting that Screen 3 included a graphic that described four components of computational thinking: decomposition, pattern recognition, algorithms, and abstraction. Participants responded to a total of seven prompts (Screens 4 through 10). The first six prompts (Screens 4-9) asked the participants to share their thoughts related to a scenario represented on the screen. The final prompt (screen 10) asked the participants to indicate whether or not they felt teaching computational thinking was important and to provide justification.

After we developed the online simulation and received approval to conduct this research from the institutional review board, faculty members at universities A, B, and C invited the preservice teachers in their courses to participate in the study and provided the students with a link to the online simulation. The preservice teachers who chose to participate accessed the simulation outside of class time and responded to its prompts. The Teacher Moments platform transcribed the participants' audio responses to the scenario prompts.

Table 2
Simulation Screens

Screen	Context
1	You have been hired to teach third grade. This is your first full-time faculty position. The school you have been hired to teach at has a lot of diversity. The students represent several different cultures, and students have varying levels of academic achievement. (read only)
2	At the first faculty meeting of the year, the principal announces a new initiative to teaching computational thinking to all students. She makes it clear that her expectation is ***all*** students will engage in learning activities that support learning computational thinking. (read only)
3	Sometimes teachers think that computational thinking is about solving math problems – but computational thinking is so much more! (Text continues by introducing the processes of computational thinking, including a graphic.) (read only)
4	As you begin to develop your lessons for teaching computational thinking, you are struggling to find appropriate lessons for your lower-achieving students. You decide to chat with the teacher in the room next to your classroom to find out what she is doing. When you ask her how she plans to teach computational thinking to her lower-achieving students, she replies, "I don't worry about it – it's not like they are going to ever need or use this information anyway." (read and record audio response)
5	You decide to talk to another teacher in the school about how she would teach computational thinking to lower-achieving students. She replies, "I just teach everybody the same – and if those students learn about computational thinking – great! And if they don't – well, they don't." (read and record audio response)
6	You decide to talk to the principal about the challenges you are feeling about creating lessons to engage your lower-achieving students in learning about computational thinking. The principal is delighted you have come to her to talk about the situation and shares, "Thank you! I am so glad you came to talk with me!" She goes on to ask you, "So what do you think might be the challenge with teaching computational thinking to lower-achieving students?" (read and record audio response)

Screen	Context
7	The principal continues to share that she knows that all students may not learn computational thinking to the same level. She states, "We need to recognize that some students are more likely to gain a deeper understanding of computational thinking than other students. However, it is important that ***ALL*** students are taught computational thinking." The principal then asks, "How might you use differentiated instruction to teach computational thinking?" How do you respond? (read and record audio response)
8	Following the discussion with the principal, you go to the faculty room to think. Another teacher enters the room and starts to chat with you. You mention you are working on ways to teaching computational thinking to all of your students. The teacher asks you why you think some students may struggle with learning computational thinking more than other students do. How do you respond to the teacher? (read and record audio response)
9	A week into the semester you are contacted by a parent of one of your lower-achieving students. The parent shares that their child is very excited to be learning computational thinking and continues to explore related topics at home. How do you respond to the parent? (read and record audio response)
10	As you think about the year that lies ahead and teaching computational thinking to all of your students, share your thoughts about whether you think it is important to teach computational thinking to ***ALL*** students, and why it is important or not. (read and record audio response)

Data Analysis

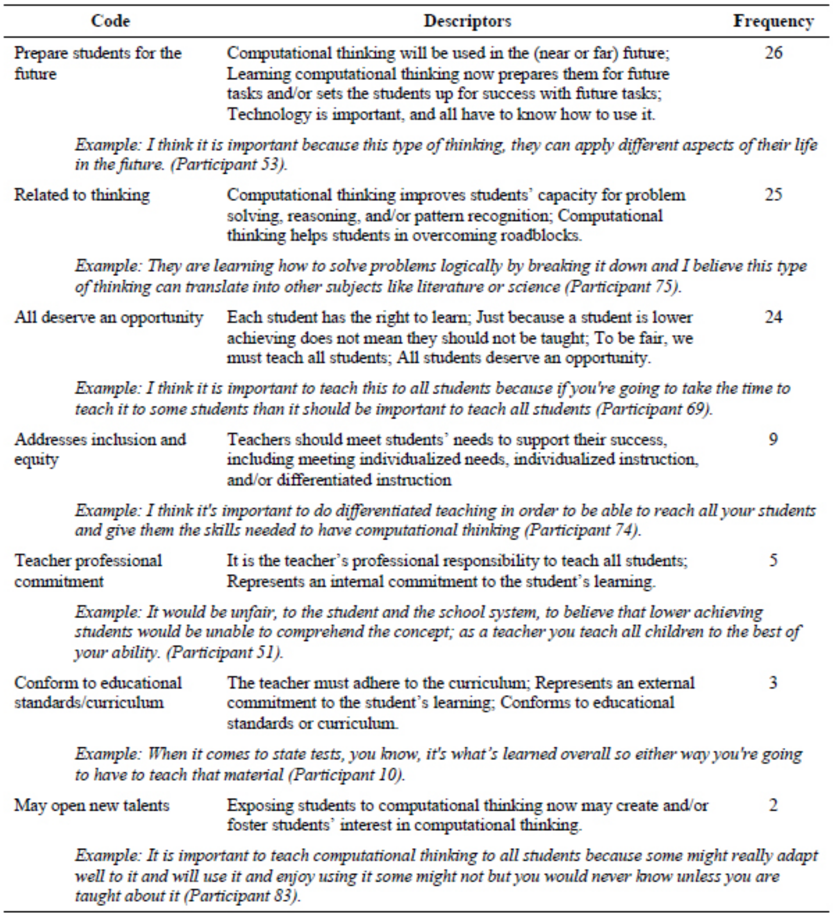
To begin the data analysis process, we developed a set of codes based on an initial reading of the responses. We discussed and defined these initial codes and then individually coded all of the responses to Screen 10's prompt regarding the importance of teaching computational thinking to all students. Participants typically responded with several thoughts to the prompt; therefore, it was possible for a response to receive multiple codes. For example, Participant 54 stated,

> YES!! All students should be provided with the opportunity to receive the same education. This means providing instruction and support to meet students where they are at academically and push them to reach high expectations and goals. While there are laws in place to make sure this happens, it is also important because we are preparing students for outside of the classroom. What students are learning leads up to future learning and, ultimately, postsecondary plans. Students need skills, such as computational thinking, to help them succeed after they graduate high school.

In this example, all three of us coded the initial statement, "All students . . .," with *All deserve an opportunity*. We also coded the phrase, "While there are laws in place to make sure this happens," with *Conform to educational standards/curriculum,* given this justification was external to the teacher. Finally, we coded the statement, "What students are learning . . .," as *Prepare students for the future*, based on the references to the support computational thinking offers for students' future learning.

While coding individually, we considered new emergent codes, as appropriate. Then, as a team we compared our codes, developed richer descriptions of the codes, discussed differences in assigned codes, and discussed the emergent codes. Afterwards, we individually recoded the complete set of responses to Screen 10, applying the updated list of seven final codes and descriptions (see Figure 1).

Figure 1
Participant Responses Regarding the Importance of Teaching Computational Thinking



Code	Descriptors	Frequency
Prepare students for the future	Computational thinking will be used in the (near or far) future; Learning computational thinking now prepares them for future tasks and/or sets the students up for success with future tasks; Technology is important, and all have to know how to use it.	26
	Example: I think it is important because this type of thinking, they can apply different aspects of their life in the future. (Participant 53).	
Related to thinking	Computational thinking improves students' capacity for problem solving, reasoning, and/or pattern recognition; Computational thinking helps students in overcoming roadblocks.	25
	Example: They are learning how to solve problems logically by breaking it down and I believe this type of thinking can translate into other subjects like literature or science (Participant 75).	
All deserve an opportunity	Each student has the right to learn; Just because a student is lower achieving does not mean they should not be taught; To be fair, we must teach all students; All students deserve an opportunity.	24
	Example: I think it is important to teach this to all students because if you're going to take the time to teach it to some students than it should be important to teach all students (Participant 69).	
Addresses inclusion and equity	Teachers should meet students' needs to support their success, including meeting individualized needs, individualized instruction, and/or differentiated instruction	9
	Example: I think it's important to do differentiated teaching in order to be able to reach all your students and give them the skills needed to have computational thinking (Participant 74).	
Teacher professional commitment	It is the teacher's professional responsibility to teach all students; Represents an internal commitment to the student's learning.	5
	Example: It would be unfair, to the student and the school system, to believe that lower achieving students would be unable to comprehend the concept; as a teacher you teach all children to the best of your ability. (Participant 51).	
Conform to educational standards/curriculum	The teacher must adhere to the curriculum; Represents an external commitment to the student's learning; Conforms to educational standards or curriculum.	3
	Example: When it comes to state tests, you know, it's what's learned overall so either way you're going to have to teach that material (Participant 10).	
May open new talents	Exposing students to computational thinking now may create and/or foster students' interest in computational thinking.	2
	Example: It is important to teach computational thinking to all students because some might really adapt well to it and will use it and enjoy using it some might not but you would never know unless you are taught about it (Participant 83).	

To compute interrater reliability for each of the seven codes, we divided the number of responses on which the coding of all three of us agreed by the total number of responses analyzed. As an example, Table 3 contains the rater data for the code, *All deserve an opportunity*. To compute the interrater reliability for this code, we divided the number of agreed upon ratings (38 + 24) by the total number of responses analyzed (80) to get a percentage of 78%. After calculating the interrater reliability for each code, the results demonstrated that, on average, we were in agreement 88% of the time, with percentages ranging from 78% to 99%.

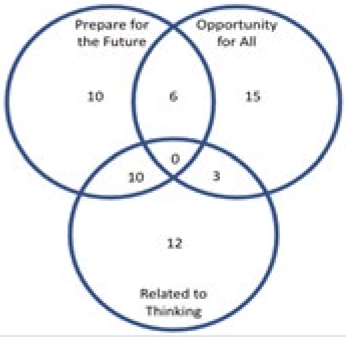
Table 3
Interrater Data for the All Deserve an Opportunity *Code*

Description	Frequency
None of the researchers coded the response with the code. (100% agreement)	38
One of the researchers coded the response with the code. (disagreement)	12
Two of the researchers coded the response with the code. (disagreement)	6
All three researchers coded the response with the code (100% agreement)	24

As demonstrated in Figure 1, we noted three codes that were assigned with relatively high frequency: *All deserve an opportunity*, *Prepare them for the future*, and *Related to thinking*. We used the responses with these codes to form three subgroups, which we analyzed further. For each participant in a subgroup, we analyzed their complete set of responses (i.e., responses to all seven prompts) looking for patterns. Three notable codes emerged: *Deficit thinking regarding low achieving students*, *Differentiated learning outcomes for students engaged in computational thinking*, and *All students can learn computational thinking*. We examined the frequencies of responses associated with these three codes across the subgroups.

In completing these analyses, it was possible for a response to receive more than one code. As a result, the *Opportunity*, *Future*, and *Thinking* subgroups were not mutually exclusive (see Figure 2). Given the amount of overlap, we examined the participant responses that were coded within a single code to determine whether the previously noted trends held.

Figure 2
Venn Diagram of Subgroups



Finally, we examined the complete responses for each participant to determine whether they were a) connecting computational thinking to the processes or components frequently mentioned in the literature, which were labeled as correct ideas or b) connecting computational thinking to mathematics classrooms (i.e., a focus on numerical computations learned in mathematics) or computers (i.e., a focus on the use of computers or technology), which were considered inappropriate ideas.

Table 4 shows the full list of codes with descriptions. Ten of the participants (12.5%) included both correct ideas and inappropriate ideas that connected to mathematics or technology. In these instances, we assigned an overall code of connecting to mathematics or connecting to technology, recognizing that the processes or components mentioned were vague and could have been offered in connection to mathematical thinking rather than computational thinking.

Table 4
Computational Thinking Connections

Code	Description	% of Participants
Correct Ideas	Participant includes processes or components that are associated with computational thinking (e.g., pattern recognition).	26
No Ideas	Participant does not connect their ideas specifically to computational thinking.	47.5
Ideas Connected to Mathematics	Participant's ideas suggest that they are focusing on computations (e.g., standard algorithm for addition) or other aspects associated with mathematics classrooms.	20
Ideas Connected to Computers/Technology	Participant's ideas are connected to the prevalence of technology in today's world and/or the importance of being able to use technology.	9

Note. The percentages exceed 100% because two participants included ideas that connected to mathematics and ideas that connected to computers/technology.

Trustworthiness

To support the validity of our research, we attended to three methodological components (Gay et al., 2012). First, we supported the evaluative validity of the research by carefully attending to our interrater reliability. Second, we practiced reflexivity by maintaining a written record of our reflections throughout the analysis process. Doing so supported the confirmability of our findings, defined as "the neutrality or objectivity of the data collected" (p. 393). Finally, we established referential adequacy by ensuring the accuracy of our interpretations of the data, which reinforced the credibility of the findings. Collectively, these actions support the trustworthiness of the study.

Limitations and Delimitations

Prior to sharing the results, it is important to consider the limitations and delimitations of the study. We noted four limitations. First, the course instructors at each university offered different incentives for participating in the study. The decision to do so was made by the individual instructors. It is not clear how the varying incentives might have influenced the participation rate or the results.

The second limitation involved participants' failure to respond to all of the prompts in the simulation. Our analysis began with the last question asked, to which three participants did not respond and were, therefore, removed from the study. Additionally, some of the 80 participants skipped questions prior to responding to that last prompt. We retained the responses of these participants in our dataset, though, as they responded to nearly all of the questions, allowing for our follow-up analyses regarding the role of equity in their responses (e.g., presence of deficit thinking). Still, it is not known what additional insights might have been drawn had they responded to all prompts.

The third limitation was related to how the participants engaged in the simulation. We did not gather any data documenting whether the participants recorded notes specifically related to the definition of computational thinking provided on Screen 3 of the simulation. Participants who provided evidence of making connections to computational thinking in their responses possibly did so without truly understanding computational thinking.

Finally, the fourth limitation involved the language used in the simulation regarding lower achieving students. It is not clear how this language might have influenced the participants who chose to use deficit-oriented language.

With regard to delimitations, we noted two. First, we drew our participants from different courses at three different universities. We based this decision on our goal of transferability, as the inclusion of participants from multiple contexts enabled the application of our findings to extend more broadly than would have been the case with participants from a single institution or within a single course. Second, we chose not to conduct follow-up interviews with participants. Although interviews might have offered additional insights, to do so would have removed the anonymity of the participants and, perhaps, influenced their responses. However, we felt the use of audio-recorded responses via the Teacher Moments platform supported our ability to gain deep insights into the participants' thoughts.

Results

Eighty participants responded to the final prompt, "Share your thoughts about whether you think it is important you teach computational thinking to ALL students. And why it is important or not." Table 3 provided the results from the analysis of these responses, including the final codes and their descriptions. Three codes emerged as primary justifications for teaching computational thinking and led to the creation of three subgroups of participants. The Opportunity subgroup included the 24 participants whose final response aligned with the code, *All deserve an opportunity*. The Future subgroup included the 26 participants whose response to the final question aligned with the code, *Prepare students for the future*. Finally, the Thinking subgroup included the 25 participants whose final response fell within the code, *Related to thinking*. The sections that follow provide descriptions and insights into these three codes and subgroups.

All Deserve an Opportunity

We coded 24 of the 80 final responses with *All deserve an opportunity*. Throughout these responses, participants communicated that all students should be afforded the opportunity to learn computational thinking regardless of individual circumstances. Half of the participants in this subgroup (*n* = 12) explicitly used the term *opportunity*. As an example, Participant 28 stated,

> I do believe that it is important to teach computational thinking to all students. It's important because each child should have the same learning opportunity as another child in the classroom. We cannot pick and choose which child gets to learn what information.

This participant focused on the need for all students to be given the same learning opportunities. In explaining this stance, they focused on the teacher, noting that the teacher should not be able to determine who is given the opportunity to learn computational thinking. In contrast, Participant 8 saw the opportunity to learn computational thinking as a student's right:

> It is very important to teach anything to everyone. Not only so everyone is on the same page and meets the standards for the year, but because education should be the one thing that is not denied to anyone based on their skills, abilities, or socio-economic status. Education is the great equalizer, or at least it should be in today's world.

By focusing on education as something that cannot be denied, the participant suggested that it is within a student's rights to learn computational thinking.

Responses related to *All deserve an opportunity* often (*n* = 10) included references to low-achieving students. For example, Participant 61 stated, "Even if they are considered low-achieving students, they still should have the opportunity to learn about computational thinking." This desire to provide low-achieving students with the opportunity to learn computational thinking was often grounded in a desire to be fair. Participant 29 stated,

> I definitely think that it is important to teach all students so that everyone has the same opportunities. And if this is helping the higher achieving students in the future, then I definitely believe that the lower achieving students should also. . . . I definitely think it's only fair and only right for them to have the opportunity to learn the same things as other students.

As shown in these examples, the participants in the Opportunity subgroup demonstrated a desire or belief that all students (rather than a select few students) should be taught computational thinking. Interestingly, we noted that these responses (n = 22) tended to focus on the idea of teaching all students without clearly addressing computational thinking, as can be seen in the previously shared response of Participant 29. The basic argument in this participant's response could have been made, for example, in regard to teaching other content, such as mathematics or social studies.

Given the general nature of teaching all students that was displayed in this subgroup's responses, we sought to gain additional insights into their ideas. To do this, we analyzed the complete set of responses (i.e., responses to all seven prompts) for each of the 24 individuals in the Opportunity subgroup. Our analysis of these responses showed that most of the participants (n = 22) felt that all students should be taught, although not all would necessarily learn. For example, Participant 2 stated,

> While it is sort of true that sometimes you are just going to get some things and sometimes you are not, you should not just say they don't get it so, you know, whatever and move on. You try to help those students that are initiating it and maybe they don't get it, but at least you will be doing all that you can to help them try to understand computational thinking and potentially understand that like their peers.

In responses like this one, we noted the deficit views of lower achieving students and the notion of different learning outcomes for students based on prior achievement. As a second example, consider the thoughts of Participant 69:

> You can split the students up into groups based on their academic level and what level they are understanding this. Then for the lower achieving levels, you can stick with the basics (what they need to know) and then just keep reviewing that, going over and not adding any other understanding or deeper levels to it. So that they just get the basics. And then for the higher achieving groups, you can start with the basics again but then go deeper as they continue to understand it more. And it may not be stuff that they have to know or anything like that. But since they are understanding it, you don't need to just keep repeating yourself about the same thing You can move on and do some more things.

Here, the participant described separating the students into different groups and then providing instruction that leads to different learning outcomes for different students. In total, 21 of the 24 participants in the Opportunity subgroup displayed deficit views of low-achieving students and 10 out of the 24 emphasized different learning outcomes.

In contrast, one participant demonstrated a commitment to differentiated instruction that supported all students in learning computational thinking. Participant 8 said,

> First, I will incorporate different methods of instruction when teaching. Secondly, I will incorporate different methods of knowledge presentation to best fit the strengths of my students. Having familiarity in instruction and knowledge presentation will relieve the stresses of trying to overcome a weakness in skill with regards to learning a certain way or presenting a certain way. This stress relief will allow the student space to learn the material, which is the ultimate goal of school. It doesn't matter so much with regards to how the material is learned, as opposed to what is learned in any degree.

In their response, Participant 8 conveyed a belief that instruction should be differentiated to meet the needs of the student so that they are successful.

In summary, our analysis revealed that 24 participants believed that students should be taught computational thinking because it is important to include all students. Their ideas focused on the unfairness of excluding students or the need to give every student a chance.

With the exception of two participants, though, this group of participants provided deficit views of lower-achieving students or described instructional practices that would lead to different learning outcomes for different groups of students. Thus, although these participants espoused views that all students deserved an opportunity to participate in instruction, they did not necessarily believe that all students could, in fact, learn computational thinking. Notably, their statements were general in nature, without making explicit connections to computational thinking.

Prepare Students for the Future

We coded 26 of the 80 final responses with *Prepare students for the future*. These participants' responses discussed how computational thinking will prepare students for the near or far future. For example, Participant 45 stated, "I feel that all students should learn at least the basics and be able to use it in their lives and in the classroom." We interpreted "use it in their lives" as referring to the distant future, whereas "in the classroom" related to the near future.

Comparably, Participant 40 stated, "Yes, I do believe it is important to teach computational thinking to all students, because when they grow up they're going to need these skills." In this response, the reference was to the far future, with emphasis on the use of computational thinking by adults. Often (n =14), responses like those of Participants 45 and 40 stated that computational thinking would be useful in the future but did so without sharing why this would be the case.

In contrast, some participants (n = 12) provided insights into why computational thinking would be beneficial in the future. For example, Participant 16 stated,

> I think that it's important to teach computational thinking to all level learners, because it is the foundation for all future learning.

> It helps build confidence in students and allows them to solve more complex equations as they move forward.

Likewise, Participant 52 stated,

> Computational thinking is important for students because it is the backbone of learning how problems, numbers and values relate to each other. Learning computational thinking will help them in their future learning and will help them achieve better in school to have a better understanding of how problems relate to each other and that there are relationships.

As shown in these responses, participants conveyed value in teaching computational thinking to students because computational thinking prepares students for future tasks and for success with future tasks.

Within these responses, two participants also connected computational thinking's future importance to the importance of understanding and working with technology. The first participant stated, "Especially in today's society, it is vital to have a basic understanding of technology and how to use it" (Participant 55). Similarly, Participant 68 stated,

> I think it is very important to teach computational thinking to students of all levels, because this is a very beneficial tool in the world nowadays. Technology is becoming very important.

These participants indicated that by learning computational thinking students would be prepared for the future, recognizing that as the importance of technology grows so will the need to understand it.

As shown in these examples, participants in the Future subgroup demonstrated a belief that by teaching students computational thinking, teachers will be preparing students for their futures. Unlike statements made by participants in the Opportunity subgroup that tended to focus on teaching, in general, we noted that these responses focused on teaching computational thinking, even if minimally, as demonstrated by Participant 52's response.

As before, the next step in the analysis was to review the complete set of responses (i.e., responses to all seven prompts) for each of the 26 participants in the Future subgroup. Similar to the results of our analysis of the complete responses from the Opportunity subgroup, the participants in the Future subgroup demonstrated deficit views of low-achieving students. For example, Participant 3 stated,

> The challenge that could be faced when teaching computational thinking to students who are lower achieving would be that they wouldn't be able to understand or comprehend such [an] intangible scenario.

Comparably, Participant 38 stated that teaching computational thinking to low-achieving students would be challenging because "students who are lower achieving [tend] to want to give up because it's more difficult for them." These notions that low-achieving students may not be capable of learning computational thinking or may not be persistent when learning computational thinking were prominent, as 21 out of the 26 participants provided statements indicating a deficit view of low-achieving students.

In contrast, eight participants in the Future subgroup indicated that all students can learn computational thinking, compared to one participant in the Opportunity subgroup. These participants in the Future subgroup believed that, if provided the right support, all students can learn computational thinking. For instance, Participant 16 stated,

> I believe that everyone should learn computational thinking no matter what level they're learning is at. Using differentiated instruction is a great way to cater to your students' needs and make sure everyone really understands the material.

Comparably, Participant 43 stated,

> I would use groups. I would find ways to reach each student in each group. They would all have different work because of their levels, but they would all be learning the same information.

These participants conveyed the belief that all students need to learn computational thinking and by understanding their students' needs a teacher can tailor instruction based on a student's understanding.

Finally, compared to the Opportunity subgroup, fewer participants in the Future subgroup (n = 4) described instructional strategies that would lead to different learning outcomes for different students. As an example, Participant 80 said,

> Knowing my students and what works best for them, I would design differentiated activities to help students in different learning positions begin to understand computation thinking. I would allow the students that really understand computational thinking to work with each other to further advance their understanding, and then I would allow the lower achieving students who might be struggling with the concept to work on a more basic activity to introduce them to computational thinking. Then after using these differentiated activities, I would mix the groups together and find an activity that's more in the middle of the two spectrums of understanding, and I'd allow those groups to try to help the students that are struggling, as well as push and promote further understanding for the students who are more familiar with computation thinking.

In summary, the analysis revealed that 26 participants believed that students should be taught computational thinking because it will prepare them for the future. Their ideas indicated that they believed understanding computational thinking would help students in the near and far future, help students with future tasks, and prepare them for the technology of the future.

Compared with the Opportunity subgroup, more participants in this Future subgroup connected, even if minimally, their ideas to computational thinking. Like the Opportunity subgroup, though, a large portion of participants in this Future subgroup demonstrated deficit views of low-achieving students. Still, more participants indicated they believed that all students could learn computational thinking, if provided the right support.

Related to Thinking

The last code that was prominently noted throughout participants' final responses was *Related to thinking*. We coded 25 out of the 80 participants' responses with *Related to thinking*. Notably, participants in this Thinking subgroup made connections to processes associated with computational thinking, thus demonstrating knowledge related specifically to computational thinking. Three processes were mentioned.

First, some participants (n = 17) shared that computational thinking would help build students' problem-solving skills. For example, Participant 37 stated, "Computational thinking is important because it can help students break down problems and look at them in other ways." Second, some participants (n = 10) indicated that computational thinking would benefit students' reasoning skills. For instance, Participant 4 stated, "I think it is important that you teach computational thinking to all students because it can help them connect the dots between content and abstract thinking."

Finally, two participants discussed how computational thinking would benefit students regarding their ability to recognize patterns. For illustration, Participant 76 stated, "I think it is important to teach all students, at least, the basics so that they can have a deeper understanding and recognition of patterns and can use those deeper thoughts in the real world." These examples demonstrate participants' views of how computational thinking can help students build their capacity for problem solving, reasoning, and pattern recognition, which are all processes associated with computational thinking.

As with the two previous subgroups, we sought additional insights into the Thinking subgroup through analysis of the complete set of responses. In this analysis, we noted patterns similar to those of the Opportunity and Future subgroups. Participants in the Thinking subgroup shared statements with deficit thinking regarding students identified as low achieving. For example, Participant 15 stated, "Lower level students might not have the level of thinking it takes to complete those problems." Similarly, Participant 45 stated,

> Computational thinking might seem unattainable because [low-achieving students] already have to learn a lot of content. If they are overwhelmed with the content [and] they're lower achievers, that might seem like just one more thing that they have to master and could become overwhelming creating more frustration.

Within the Thinking subgroup, 18 out of 25 participants shared deficit thinking regarding low-achieving students learning computational thinking.

Beyond this deficit thinking, participants in the Thinking subgroup described differentiating instruction in ways that would lead to differentiated outcomes. For example, Participant 32 stated, “Everyone shouldn’t be taught the same, students require different expectations based on their ability levels.” Similarly, Participant 36 stated,

> I could have different groups based on different levels of learners. So one group may be higher learners, so I'll have them do more with the computational thinking. And then my lower-level groups will still use that process, but it will not be to such an extent as my higher level.

A total of 6 participants in the Thinking subgroup described instruction in this way.

In contrast, other participants in the Thinking subgroup shared that all students can learn computational thinking provided the right support. For instance, Participant 15 stated, “The lower level students [will] learn how computational thinking works on their level, and then we [will] set up harder problems as they learn to do it.” Similarly, Participant 59 stated,

> Students can decide whichever way they choose to learn the material so that they can come out of this successfully. Using different types of teaching methods allows all students to learn just in different ways. Every student learns differently, and it is important to allow them to choose.

When looking at the complete set of responses of the 25 participants whose responses were a part of the Thinking subgroup, six participants’ responses shared ideas that aligned with the belief that all students can learn computational thinking.

In summary, 25 participants’ final responses aligned with the code, *Related to thinking*. These participants discussed how computational thinking could build students’ skills in problem solving, reasoning, and pattern recognition and, in doing so, provided evidence of knowing processes associated with computational thinking. Even though the participants discussed how beneficial they believed learning computational thinking would be for students, they shared deficit ideas regarding low-achieving students and their ability to engage in computational thinking. Some participants believed that only some students could understand computational thinking to the highest degree, but others believed that all could eventually have a deep understanding of computational thinking when given the right support.

Subgroup Comparisons

Our initial analysis of the 80 participants’ responses revealed three dominant codes regarding the importance of teaching computational thinking. Responses within each of these three codes were further analyzed. Table 5 provides a summary of these results.

Table 5
Participant Count by Code

Code	Deficit Perspective	Differentiated Outcomes	All Students Can Learn CT
Opportunity for All (*n* = 24)	21 (88%)	10 (42%)	1 (4%)
Prepares Them for the Future (*n* = 26)	21 (81%)	4 (15%)	8 (31%)
Related to Thinking (*n* = 25)	18 (72%)	6 (24%)	6 (24%)

A large majority of participants in all three subgroups demonstrated a deficit perspective of low-achieving students. Instruction that would enable different students to learn computational thinking to varying degrees (i.e., differentiated outcomes) was more prevalent in the Opportunity subgroup compared to the others. Conversely, when compared to the Opportunity subgroup, a higher percentage of participants in the Future and Thinking subgroups believed that all students can successfully learn computational thinking.

Recognizing that the three subgroups were not mutually exclusive, we examined the participant responses that were coded with a single code (see Table 6) to determine whether the previously noted trends held. In general, these trends were noted among the mutually exclusive groups, with the exception of the reduced belief in the Thinking subgroup that all students can learn computational thinking.

Table 6
Participant Count by Code – Mutually Exclusive Groups

Code	Deficit Perspective	Differentiated Outcomes	All Students Can Learn CT
Opportunity for All (*n* = 15)	13 (87%)	5 (33%)	0 (0%)
Prepares Them for the Future (*n* = 10)	7 (70%)	0 (0%)	5 (50%)
Related to Thinking (*n* = 12)	9 (75%)	3 (25%)	1 (8%)

Connections to Computational Thinking

Our final analysis involved an examination of the connections participants made to computational thinking in their responses. As demonstrated in Table 4, nearly half of the participants completed the simulation by sharing about teaching, in general, without specifically referencing processes or components of computational thinking. In contrast, 26% of participants included processes or components associated with computational thinking. These participants typically spoke about problem solving or breaking down problems.

Three participants in this group specifically listed the four processes of computational thinking featured on the third screen of the simulation (i.e., decomposition, pattern recognizing, algorithms, and abstraction). The remaining participants offered insights that connected to mathematics classrooms (20%) and/or technology integration (9%).

Discussion and Conclusion

With the growing emphasis on computing, students must be prepared to use technology as a problem-solving tool (NCTM, 2014; Sykora, 2021). As a result, ISTE (2021, ISTE & CSTA, 2011) has called for the inclusion of computational thinking across all disciplines in the K-12 curriculum. Notably, mathematics classrooms appear to be an obvious place to advance computational thinking given their emphasis on problem solving, problem posing, and mathematical modeling (Gadanidis et al., 2017; Kallia et al., 2021). It is important, then, for mathematics teacher educators to consider the perspectives of preservice teachers who will likely be called upon to include computational thinking in their mathematics lessons and yet who have little to no background in computational thinking.

Previous studies have been limited to relatively small samples (e.g., Bower & Falkner, 2015) and often drawn from single university settings (see also, Gadanidis et al., 2017). In contrast, the current study had a large sample (n = 80) drawn from three universities. Therefore, the current study informs and strengthens the field's burgeoning understanding of preservice teachers' perspectives related to computational thinking.

The results of the current study demonstrated that the participants, in general, thought that it was important to teach computational thinking to elementary students and justified this assertion with three primary reasons. First, participants believed all students should have the opportunity to learn computational thinking. In some ways, these responses aligned with those in Gadanidis et al.'s (2017) study that focused on the integration of computational thinking and mathematics. Gadanidis et al.'s participants saw value in the integration of mathematics and computational thinking and, as a result, perceived students should have the opportunity to learn this integrated content.

In contrast, participants in the current study held a general belief that all students should have an opportunity to learn regardless of the topic being taught. Our participants did not necessarily connect the opportunity to learn with the value of computational thinking, possibly due to not having participated in course experiences focused on computational thinking.

The second justification involved participants' stance that learning computational thinking prepares students for tasks in a technology-driven future. Gadanidis et al. (2017) coded a similar set of responses with the code *Computational Thinking and Society*. It appeared that in both studies, participants considered technology or computers as a tool that will be leveraged for solving problems in the future. Interestingly, some definitions of computational thinking focus on thinking processes that lead to algorithms without specifying the use of the computer to code and carry out the algorithms (cf., Edwards & Cassidy, 2021). It is not clear how participants' views might change when considering computational thinking in contexts that do not require a computer.

The third justification was related to the role of computational thinking in developing capacity for problem solving and general thinking processes. Like participants in Bower and Falkner's (2015) study, the participants in the current study did not demonstrate a deep understanding of computational thinking. Within the sample, almost half of the participants did not connect their responses to computational thinking, providing responses that could have been applied to many different content areas. Other participants erroneously connected their ideas to numerical computations, typically considered a component of mathematics classes, or general technology integration.

These erroneous connections confirmed the predictions made by Edwards and Cassidy (2021). Interestingly, the small group of participants (n = 21) that appropriately connected their ideas to computational thinking did so by identifying processes or components associated with computational thinking that are shared with mathematical thinking. As a result, we could not determine whether these participants would be able to distinguish between computational thinking and mathematical thinking. We anticipate, though, that the participants' general sense of the processes associated with computational thinking were likely not as rich as those of preservice teachers who had completed a course on computational thinking (Gadanidis et al., 2017).

Unique to the current study was our examination of participants' views through an equity lens. Throughout the justifications for the importance of teaching computational thinking, a large majority of participants shared ideas that communicated not all students were capable of learning computational thinking.

Further, they talked about differentiated instruction as a means for giving easier problems or tasks to lower-achieving students rather than as a tool to scaffold instruction and support all students in meeting targeted learning goals. Only a small number of participants indicated that all students are capable of learning computational thinking. It was unclear whether these deficit views were specific to the teaching and learning of computational thinking, were specific to the scenario represented in our simulation, or represented general views of teaching and learning.

Implications

The results of our equity-focused analysis lead to three implications. First, without the inclusion of computational thinking in their teacher preparation programs, many participants did not justify teaching computational thinking with reasons that were actually connected to the benefits of learning computational thinking. When connections to computational thinking were made, they were superficial. In this way, our results add to the field's understanding of preservice teachers' perspectives of computational thinking. To strengthen this understanding, though, future research should provide preservice teachers with the opportunity to communicate their understanding of computational thinking, perhaps through stating explicit definitions or providing example tasks or lessons aimed at developing computational thinking.

Our results also confirm the need for preservice teachers to explicitly and strategically be engaged in learning computational thinking content and pedagogy within their teacher preparation programs (Barr & Stephenson, 2011; Bower & Falkner, 2015). We acknowledge, though, that doing so will most likely not be as simple as adding an additional course to programs, and we call on teacher educators to work collaboratively to address this complex issue.

Second, when participants made connections to computational thinking in their responses, they either erroneously connected to numerical computations or technology integration, or they limited their responses to processes that are shared with mathematical thinking. As a result, in preparing preservice teachers, mathematics teacher educators should include opportunities to examine the commonalities and differences between mathematical thinking and computational thinking. Future research should consider how understanding the divergent nature of mathematical thinking and computational thinking might serve to enhance preservice teachers' understanding and ability to plan and effectively teach these two subjects.

Finally, when viewed through an equity lens, our results confirm the need for teacher preparation programs to explicitly attend to issues of equity in teaching computational thinking, a need that has been recently expressed by ISTE (2021). Future work should examine the reasoning behind preservice teachers' equity-related views. In this work, researchers should seek to delineate preservice teachers' vision of high-quality teaching, in general, as well as their vision of computational thinking, specifically. In addition, it is important to recognize how understanding computational thinking or engaging in computational thinking influences a preservice teacher's interests in advancing equity in teaching computational thinking. Understanding preservice teachers' visions of an ideal classroom, whether equitable or inequitable, will serve to inform teacher preparation programs.

In addition to these implications that follow from our results, our methodological design also leads to an implication of the study. To collect our data, we used the Teacher Moments platform, a tool that allows participants to provide audio recordings of their ideas in response to written prompts and scenarios. This online, asynchronous simulation platform was designed as an instructional tool to support teachers practicing their responses to students in classroom situations (Teaching Systems Lab, 2020). In contrast, we utilized the platform as a research tool. We leveraged the platform to impartially collect audio responses from a large number of participants and to transcribe the audio data. As a result, this study demonstrated the utility of the Teacher Moments platform for data collection.

Conclusion

With the increasing interest in integrating computational thinking across the K-12 curriculum (Kafai & Proctor, 2021), researchers have acknowledged the challenge of preparing teachers to effectively teach computational thinking (Cuny, 2011). Mathematics classrooms will likely be selected as a context for including computational thinking (Gadanidis et al., 2017; Kallia et al., 2021), despite the issues computational thinking presents in learning mathematics (Bråting & Kilhamn, 2021; Rich et al., 2020). It is important, then, for mathematics teacher educators to consider preservice teachers' perspectives as they contemplate how to include computational thinking within the teacher preparation programs. The results of this study serve not only to inform those who are designing teacher preparation experiences for teaching computational thinking but also to guide the field with additional areas to consider in future work.

References

Association of Mathematics Teacher Educators. (2017). *Standards for preparing teachers of mathematics*. https://amte.net/standards

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54.

Battey, D., & Stark, M. (2009). Inequitable classroom practices: Avoiding ability attributions from misconceptions in mathematics. In C. Malloy (Ed.), *Mathematics for all: Instructional strategies for diverse classrooms* (pp. 167–177). National Council of Teachers of Mathematics.

Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning, 23*(2), 170-185.

Bouck, E. C., Sands, P., Long, H., & Yadav, A. (2021). Preparing special education preservice teachers to teach computational thinking and computer science in mathematics. *Teacher Education and Special Education*, *44*(3), 221-238.

Bower, M., & Falkner, K. (2015). Computational thinking, the notional machine, pre-service teachers, and research opportunities. In D. D'Souza & K. Falkner (Eds.), *Proceedings of the 17th Australasian Computing Education Conference* (pp. 37–46). Australian Computer Society Inc.

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

Chang, Y. H., & Peterson, L. (2018). Pre-service teachers' perceptions of computational thinking. *Journal of Technology and Teacher Education*, *26*(3), 353-374.

Corbin, J., & Strauss, A. (2008). *Basics of qualitative research* (3rd ed.). Sage.

Creswell, J. W. (2013). *Qualitative inquiry a nd research design: Choosing among five approaches* (3rd ed.). Sage.

Cuny, J. (2011). Transforming computer science education in high schools. *Computer, 44*(6), 107–109.

Denning, P. J. (2005). Is computer science science? *Communications of the ACM*, *48*(4), 27-31.

Diversity in Mathematics Education Center for Learning and Teaching. (2007). Culture, race, power, and mathematics education. In F. K. Lester, Jr. (Ed.), *Second handbook of research on mathematics teaching and learning* (pp. 405–433). National Council of Teachers of Mathematics and Information Age Publishing.

Edwards, T., & Cassidy, M. (2021). *Exploring computational thinking: A look into the different definitions and understandings of computational thinking*. TERC.

Fessakis, G., & Prantsoudi, S. (2019). Computer science teachers' perceptions, beliefs, and attitudes on computational thinking in Greece. *Informatics in Education, 18*(2), 227–258.

Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). Computational thinking in mathematics teacher education. *Contemporary Issues in Technology and Teacher Education, 17*(4), 458–477. https://citejournal.org/volume-17/issue-4-17/mathematics/computational-thinking-in-mathematics-teacher-education

Gay, L. R., Mills, G. E., & Airasian, P. (2012). *Educational research: Competencies for analysis and applications* (10th ed.). Pearson.

Hammerness, K. (2003). Learning to hope, or hoping to learn? The role of vision in the early professional lives of teachers. *Journal of Teacher Education, 54*(1)*,* 43–56.

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296-310.

International Society for Technology in Education. (2021). *Computational thinking competencies*. https://www.iste.org/standards/computational-thinking

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

Kafai, Y. B., & Proctor, C. (2021). A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educational Researcher, 51*(2), 146-151.

Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education, 23*(2), 159–187.

Martin, D. B., Anderson, C. R., & Shah, N. (2017). Race and mathematics education. In J. Cai (Ed.), *Compendium for research in mathematics education* (pp. 607–636). National Council of Teachers of Mathematics.

Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education, 19*(4), 790-824. https://citejournal.org/volume-19/issue-4-19/general/preparing-elementary-school-teachers-to-teach-computing-coding-and-computational-thinking

Milner, H. R. (2012). Beyond a test score: Explaining opportunity gaps in educational practice. *Journal of Black Studies, 43*(6), 863–894.

Monteleone, C., White, P., & Geiger, V. (2018). Defining the characteristics of critical mathematical thinking. In J. Hunter, P. Perger, & L. Darragh (Eds.), *Making waves, opening spaces* (pp. 559-566). MERGA.

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).

National Association for the Education of Young Children. (2019). *Advancing equity in early childhood education*. https://www.naeyc.org/resources/position-statements/equity

National Council of Teachers of Mathematics. (2014). *Principles to actions: Ensuring mathematical success for all.* Author.

National Governors Association Center for Best Practices & Council of Chief State School Officers. (2010). *Common core state standards for mathematics*. https://learning.ccsso.org/wp-content/uploads/2022/11/Math_Standards1.pdf

National Research Council. (2001). *Adding it up: Helping children learn mathematics*. National Academies Press.

National Research Council. (2010). *The scope and nature of computational thinking*. National Academies Press.

Polard, T. M. (2013). Unpacking everyday "teacher talk" about students and families of color: Implications for teacher and school leader development. *Urban Education, 48*(6), 863–894.

Rich, K. M., Spaepen, E., Strickland, C., & Moran, C. (2020) Synergies and differences in mathematical and computational thinking: Implications for integrated instruction. *Interactive Learning Environments, 28*(3), 272-283.

Rycraft-Smith, L., & Connolly, C. (2019). Comparing conceptions of mathematical and computational thinking cycles. *Espresso, 24*. https://www.cambridgemaths.org/Images/espresso_24_mathematical_and_computational_thinking.pdf

Shulman, L. S., & Shulman, J. H. (2004). How and what teachers learn: A shifting perspective. *Journal of Curriculum Studies, 36*(2)*,* 257–271.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158.

Sleeter, C. E. (2008). Preparing White teachers for diverse students. In M. Cochran-Smith, S. Feiman-Nemser, D. J. McIntyre, & K. E. Demers (Eds.), *Handbook of research on teacher education: Enduring questions in changing contexts* (3rd ed.; pp. 559–582). Routledge/Taylor & Francis group and the Association of Teacher Educators.

Sneider, C., Stephenson, C., Schafer, B., & Flick, L. (2014). Exploring the science framework and NGSS: Computational thinking in the science classroom. *Science Scope*, *38*(3), 10-15.

Sykora, C. (2021). *Computational thinking for all*. International Society for Technology in Education. https://www.iste.org/explore/computational-thinking/computational-thinking-all

Tall, D. (1991). The psychology of advanced mathematical thinking. In D. Tall (Ed.), *Advanced mathematical thinking* (pp. 3–21). Kluwer Academic.

Teaching Systems Lab. (2020). *Teacher moments*. https://tsl.mit.edu/practice_space/teacher-moments/

Turner, R., & Angius, N. (2017). The philosophy of computer science. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy (Summer 2017 ed.)*. https://plato.stanford.edu/archives/sum2017/entries/computer-science/

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.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. In P. J. Rich & C. B. Hodges (Eds). *Emerging research, practice, and policy on computational thinking* (pp. 205-220). Springer.

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.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 80*(4), 55-62.