

Achieving CSforAll: Preparing Special Education Pre-service Teachers to Bring Computing to Students with Disabilities

Aman Yadav
Michigan State University
East Lansing, MI, USA
ayadav@msu.edu

Maya Israel
University of Florida
Gainesville, FL, USA
misrael@coe.ufl.edu

Emily Bouck
Michigan State University
East Lansing, MI, USA
ecb@msu.edu

Alexis Cobo
University of Florida
Gainesville, FL, USA
awolfson@ufl.edu

John Samuels
University of Florida
Gainesville, FL, USA
jsamuels1@ufl.edu

ABSTRACT

While computational thinking has gained popularity in K-12 schools to increase access to computing tools and practices, there is still limited understanding on how to broaden participation of students with disabilities in computational thinking (CT). One approach to increasing access to computing to students with disabilities is to educate future special education teachers to bring CT into their instruction. This study examined the influence of integrating CT into assistive technology course for special education pre-service teachers. Our results suggest that integrating CT into special education teacher preparation coursework can have a positive impact on how pre-service teachers see the value of bringing computational practices to students with disabilities.

CCS CONCEPTS

• **Social and professional topics** → **Computer science education**.

KEYWORDS

Computational Thinking, Pre-service Teachers

ACM Reference Format:

Aman Yadav, Maya Israel, Emily Bouck, Alexis Cobo, and John Samuels. 2022. Achieving CSforAll: Preparing Special Education Pre-service Teachers to Bring Computing to Students with Disabilities. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022)*, March 3–5, 2022, Providence, RI, USA. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3478431.3499333>

1 INTRODUCTION

While Wing popularized CT [16], the idea of using computers to solve problems or “thinking with a computer” (Papert, 1993) go as far back as 1960s when Alan Perlis argued that “designing algorithms to solve problems involved basic thought processes that

everyone should eventually learn” [3]. Wing argued CT is a fundamental skill of analytical thinking for everyone, not just computer scientists. She described computational thinking (CT) as “solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” [16]. Wing also pointed out the untapped potential of CT for K-12 education by stating, “To reading, writing, and arithmetic, we should add computational thinking to every child’s analytical ability.” A report on computational thinking by the National Council for Research advanced a similar idea that CT is a cognitive skill that “average person is expected to possess.” The National Research Council report highlighted “(1) that students can learn thinking strategies such as computational thinking as they study a discipline, (2) that teachers and curricula can model these strategies for students, and (3) that appropriate guidance can enable students to learn to use these strategies independently” [5]. Recent educational reforms in K-12 education, such as the Next Generation Science Standards (NGSS) and Common Core State Standards (CCSS), either explicitly or implicitly also called out the need for students to engage in computational tools and practices (Yadav, Hong, Stephenson, 2016). A report by Digital Promise (2019) argued CT is “important to know and know how to do it in a computational world” [12]. Computational thinking in education has the potential to significantly advance computing skills and ability of K-12 students as they begin to think and problem-solve in new ways. In one study, Lewandowski and colleagues illustrated the idea of the “common-sense programming” found in non-programming students. Students were asked to propose solutions to avoid selling the same seat twice at multiple box offices for a theatre [9]. The results showed that 69% of the solutions obtained were correct, which indicated that the non-programming students are equipped with the natural understanding of some basic concepts to solve problems computationally, such as the “concurrency” concept applied in this case. Integrating computational thinking in the K-12 curriculum would certainly enhance students’ ability to abstract concepts and increase their problem-solving skills in core content areas.

In recent years, computational thinking has gained tremendous popularity in K-12 and has primarily been driven by professional learning opportunities for in-service teachers. For example, in one study Rich, Yadav, and Schwarz found that elementary teachers see natural connections between CT and their classroom activities [14]. In particular, teachers focused on the problem-solving aspects of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGCSE 2022, March 3–5, 2022, Providence, RI, USA

© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9070-5/22/03...\$15.00
<https://doi.org/10.1145/3478431.3499333>

CT and made stronger connections between CT and mathematics than between CT and science. The authors also found teachers were able to make some connections between their classroom practices and each of the six CT practices presented to them: abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization. As CT continues to gain popularity in K-12 classrooms through in-service teacher professional development, we also need to focus on preparing future teachers in integrating computational thinking in their future teaching.

2 COMPUTATIONAL THINKING IN TEACHER EDUCATION

There has been emerging research on integrating CT in general teacher education courses, such as Introduction to Educational Technology and Introduction to Educational Psychology courses. For example, Yadav and colleagues examined the influence of the two-week module in an Introduction to Educational Psychology course on pre-service teachers' understanding of CT and attitudes toward integrating CT in their future classrooms [17]. The authors found the module significantly influenced pre-service teachers' understanding of CT as they developed a more nuanced view of CT and also positively influenced their attitudes towards bringing CT into teaching.

In another study, Mouza et al. redesigned an educational technology course to introduce computing tools, vocabulary, and practices to pre-service teachers to develop content and pedagogical knowledge to integrate CT in K-8 classrooms[10]. The authors used a survey to measure pre-service teachers' understanding and dispositions towards CT and case studies that pre-service teachers developed after designing, implementing, and reflecting on their own lessons to support the development of CT in students. Researchers found integrating CT in an educational technology course can positively impact pre-service teachers' attitudes toward CT as well as improve their understanding of CT concepts. In addition, the analysis of cases suggested that pre-service teachers recognised the interacting relationships or fit among disciplinary content, pedagogy, and computational tools that support CT knowledge and skills. Similarly, Chang and Peterson redesigned an educational technology course around CT and asked pre-service teachers in the course to reflect on their CT learning after the in-class CT activities using open-ended questions. The authors concluded pre-service teachers' connected CT to their own educational philosophy and how students could show their computational skills through projects. In addition, participants developed positive attitudes towards CT.

While there is emerging evidence of how CT influences pre-service teachers' understanding and attitudes, it is still primarily limited to general education courses. Specifically, the majority of CT integration work has focused on preparing pre-service teachers for general education settings and there is no work we are aware of that prepares pre-service teachers to use CT in a special education context to support students with disabilities. Given the goals of *CSforAll* movement include bringing computing to all K-12 learners including students with disabilities, we need to also prepare special education teachers to integrate CT into their instruction.

3 COMPUTATIONAL THINKING IN SPECIAL EDUCATION

There is emerging research on bringing computational experiences for students with disabilities with teachers and administrators reporting student's disability status as a barrier to full participation in computing [8]. One means to address the barriers is Universal Design for Learning (UDL), which is a framework to "to improve and optimize teaching and learning for all people based on scientific insights into how humans learn" [1]. UDL includes three guidelines for learning: (i) providing multiple means of engagement in learning, (ii) providing multiple means of representation of content, and (iii) providing multiple means for students to express their understanding.

Research on UDL in computing context has only been conducted with in-service teachers and how they believe UDL can support students with disabilities in learning. In one study on school-wide implementation of CT, researchers observed that teachers already use some instructional strategies, such as modeling, explicit instruction, and peer collaboration to support struggling learners in computing [8]; however, teachers use of UDL strategies is limited [13]. In another study, Israel and colleagues examined how elementary teachers implement UDL strategies for an academically diverse group of students in the context of CS [7]. The authors found teachers taught in ways that was consistent with the UDL framework, but their implementation was narrow as they emphasized some strategies over others. In addition, teachers' use of UDL within CS context was not any different than other content areas. The authors reported teachers needed significantly more support and professional development from a coach familiar with both CS and UDL during the planning as well as implementation.

While the work on bringing CT to in-service teachers to engage students with disabilities in computing is emerging, there are few studies on how to prepare special education pre-service teachers in both CT as well as using UDL for teaching CT. There is one study that we are aware of that examined how special education pre-service teachers integrate CT in their mathematics lessons [2]. In this study, researchers examined how exposure to CT/CS content influences the integration of CT concepts into mathematics lesson plans developed by the special pre-service teachers for students with disabilities. The authors analyzed lesson plans from 31 pre-service teachers using a rubric that focused on both CT concepts, math concepts, and integration of CT into math. Additional data included reflections on the lessons to measure pre-service teachers' perceptions about CT and confidence in implementing CT. The results suggest that overall special education pre-service teachers were able to integrate CT into their lessons and were positive towards CT; however, they need additional exposure to computing and experiences with integrating CT/CS into mathematics lessons. Pre-service teachers in this study did not use UDL to support students with disabilities in computing.

Given the limited work on how to prepare special education pre-service teachers to use UDL to bring CT into their future classrooms, our goal in this study was to examine how infusing CT in an assistive technology course influences special education pre-service teachers. Specifically, our research was guided by the following questions:

- (1) What is the influence of exposure to computational thinking on special education pre-service teachers' attitudes towards computing for students with disabilities?
- (2) How do pre-service teachers use Universal Design for Learning (UDL) principles to support students with disabilities in computing?

4 METHODS

4.1 Context

This study was conducted in an assistive technology course within a special education teacher preparation program at a large Mid-western University. The course goal focused on understanding how assistive technologies can improve, maintain, or increase the academic and functional capabilities of students with disabilities in K-12 education. The course was 14 weeks in length and occurred virtually due to the COVID-19 pandemic. We introduced computational thinking practices and tools during two weeks (Week 10 and Week 11) of the course. The first CT week involved pre-service teachers reading two articles on computational thinking, exploring Scratch programming environment, and exploring CT practices (algorithms, debugging, decomposition, abstraction, and pattern matching) through unplugged activities. The second week of CT included exploring Scratch to teach geometrical shapes, readings on UDL within the context of programming and CT, and applying UDL principles to a CT lesson.

4.2 Participants

Seventeen special education pre-service teachers complete all aspects of the study. Participants included 16 women and 1 man, which is typical representation for teacher education programs including special education [11]. Three pre-service teachers were focused on secondary education and 14 were focused on elementary education.

4.3 Measures

Survey.

In order to measure the impact of integrating computational thinking and computer science on pre-service teachers we adapted the Teacher Beliefs about Coding and Computational Thinking (TBaCCT) instrument [15]. We changed the items from a focus on elementary students to students with disabilities. For example, we changed the item "Learning about computing can help elementary students become more engaged in school" to "Learning about computing can help students with learning disabilities become more engaged in school".

TBaCCT includes three specific beliefs relevant to teaching computational thinking and coding to young children: value beliefs, self-efficacy beliefs, and teacher efficacy beliefs. The value beliefs measured the importance teachers placed on computing education for students, the self-efficacy beliefs measured pre-service teachers' own efficacy in CT, and the teaching efficacy beliefs measured pre-service teachers' efficacy for teaching computing. In addition, we added four Likert-scale items that measured pre-service teachers' beliefs about using UDL to teach computing to support students with disabilities.

Case Study Scenario

In order to measure how pre-service teachers apply UDL principles to support students with disabilities in computing experiences, we developed a case study that presented a classroom scenario where pre-service teachers were co-planning and co-teaching a CT lesson that used Scratch to teach geometric shapes. The scenario presented three hypothetical students with different disabilities and asked pre-service teachers how the three UDL principles (Multiple Means of Engagement, Multiple Means of Representation, Multiple Means of Action Expression) could be applied in the lesson. Pre-service teachers responded to six open-ended questions about what UDL feature they noticed and what UDL feature they would add for each of the three principles. Appendix A includes the scenario.

4.4 Data Collection and Analysis

Pre-service teachers enrolled in the assistive technology course were asked to complete a pre-survey about 2 weeks before the CT modules were implemented in the course. They received the post-survey after the CT modules. We sent two reminders to the pre-service teachers to complete the pre-service and the post-survey.

The survey items under each of the sub-sections were combined to give a composite score for pre-service teachers' value beliefs, teaching efficacy in computational thinking, efficacy of their own computational thinking knowledge, and using UDL strategies to support students with disabilities in computational thinking. We analyzed the complete scores using the Wilcoxon Signed Rank Test to check if pre-service teachers' beliefs were significantly different between before and after the CT integration in the course. The Wilcoxon Signed Rank test is similar to dependent paired t-test, but does not assume that data is normal.

The open-ended responses to the case study scenario were analyzed using the UDL guidelines for computer science and computational thinking framework. The framework provides concrete strategies for using the three UDL principles within the context of CS/CT lessons. The goal of the analysis was to identify which UDL strategies pre-service teachers use to support students in CT/CS learning experiences. In order to develop a coding scheme, two coders independently applied the framework to four open-ended responses for each UDL principle and discussed any discrepancies to ensure they were applying the framework appropriately. Once the two coders had established an understanding of how to code the data using the framework, they independently coded 20% of the open-ended responses to establish inter-rater reliability. We used Maxwell RE to calculate the reliability, which was 0.84 and suggested a strong inter-rater agreement. The two coders, then, split the remaining data and coded independently.

5 RESULTS

Descriptive statistics from the survey indicated that overall the integration of computational thinking in assistive technology course increased pre-service teachers' value beliefs, teaching efficacy, efficacy in CT, and UDL strategies. Table 1 shows the pre and post means and standard deviations for the four survey sections.

A Wilcoxon Signed-Ranks test indicated that pre-service teachers' value beliefs were significantly higher after the intervention (mean rank = 8.82) than before (mean rank = 9.83), $Z = -2.23$, $p = 0.03$. Results from the Wilcoxon Signed-Ranks test also indicated

Table 1: Descriptive Statistics

	Pre Mean (SD)	Post Mean (SD)
Value Beliefs	46.06 (5.66)	49.65 (7.06)
Teaching Efficacy	37.53 (10.11)	45.88 (5.67)
CT Efficacy	11.35(1.96)	12.18 (2.20)
UDL	15.64 (2.52)	19.96 (3.10)

that pre-service teachers' own efficacy beliefs about teaching computing were significantly higher after the intervention (mean rank = 9.61) than before (mean rank = 6.17), $Z = -2.75$, $p = 0.006$. Wilcoxon Signed-Ranks also test indicated that pre-service teachers' beliefs about using UDL strategies to support students with disabilities in computing were significantly higher after the intervention (mean rank = 9.14) than before (mean rank = 4.00), $Z = -3.11$, $p = 0.002$. It is interesting to note that pre-service teachers' own self-efficacy in CT was NOT significantly different between before and after the intervention (mean rank = 8.06) than before (mean rank = 6.50), $Z = -1.27$, $p = 0.20$.

Analysis of the open-ended responses suggested that pre-service teachers' use of UDL principles primarily focused on multiple means of engagement with 143 codes emerging from the data. Multiple means of engagement principle focuses on providing student with multiple ways to engage in the content given that one way is not ideal for all learners and include three guidelines: (a) Recruiting Interest (spark excitement and curiosity for learning); (b) Sustaining Effort Persistence (Tackle challenges with focus and determination.); and (c) Self Regulation (Harness the power of emotions and motivation in learning) [1]. Table 2 shows the number of codes that emerged for each of the guidelines and example response.

The open-ended responses showed that pre-service teachers in our study discussed multiple means of engagement most to support students with disabilities. Within this principle, sustaining effort was the most suggested strategy. In order to ensure that students had sustained effort during CT/CS learning experiences, pre-service teachers suggested using group work or pair programming. Recruiting interest was the next UDL strategy pre-service teachers suggested as they focused on providing students with choices within CT/CS activities.

For multiple means of representation, there were 76 codes that emerged from the data. Multiple means of representation focuses on providing more than one means of representing information given that learners perceive and comprehend information in different ways. This principles include three guidelines: (a) Perception (Interact with flexible content that doesn't depend on a single sense like sight, hearing, movement, or touch); (b) Language and Symbols (Communicate through languages that create a shared understanding); and (c) Comprehension (Construct meaning and generate new understandings) [1]. Table 3 shows the number of codes that emerged for each of the guidelines and example response.

Pre-service teachers in our study had the least number of responses that focused on multiple means of representation. Under this principle, perception had the highest number of codes that included providing students with instructions on how to accomplish the task using a video. Comprehension had the second most codes

Table 2: Multiple Means of Engagement

Guidelines	N = 143	Example
Recruiting Interest	41	This lesson allows for individualization through scratch. The students can change sprites and customize the movements of their sprites.
Sustaining Effort	77	I would also create many break opportunities where we take breaks from the code and discuss progress as a whole group to prevent Connie from getting too overwhelmed and frustrated.
Self Regulation	25	One feature I noticed is communicating clear expectations for computing tasks, collaboration, and help seeking

Table 3: Multiple Means of Representation

Guidelines	N = 76	Example
Perception	35	She can have instruction or computing skills read aloud to her or provide her with a video to show her the steps of tasks
Language and Symbols	16	Post anchor charts and provide reference sheets with images of blocks or with common syntax when using text
Comprehension	25	Encourage students to ask questions as comprehension checkpoints

for this principle as participants responded that using questions to check for understanding. Language and Symbols was the last category that emerged from the responses and included providing anchor charts in the classroom that students could use as references when coding.

For multiple means of action and expression, there were 114 codes that emerged from the data. Multiple means of action and expression is about how learners navigate their learning environment and includes three guidelines: (1) Physical Action (Interact with accessible materials and tools); (b) Expression and Communication (Compose and share ideas using tools that help attain learning goals); (c) Executive Functions (Develop and act on plans to make the most out of learning) [1]. Table 4 shows the number of codes that emerged for each of the guidelines and example response.

Multiple means of action and expression was the second most UDL principle that pre-service teachers in our study discussed in their responses. Within this principle, physical action had the highest number of codes with participants discussing physical assistive technologies for students with disabilities. Expression and Communication emerged as the second most coded strategy under

Table 4: Multiple Means of Action and Expression

Guidelines	N=114	Example
Physical Action	61	Robert would benefit from using Scratch on a touch screen device and using a stylus to select the coding blocks.
Expression and Communication	27	Creating physical manipulatives for the students to use as coding blocks
Executive Functions	26	Provide graphic organizers to facilitate planning, goal-setting, and debugging

this principle as participants suggested using physical manipulatives such as cutout physical coding blocks. Finally, Executive functioning was coded 26 times with pre-service teachers focusing on providing explicit instruction to students in help seeking and general problem solving techniques.

6 DISCUSSION

Results from our study show that as a result of introducing computational thinking and coding to special education pre-service teachers they saw it as being relevant to students with disabilities. In addition, pre-service teachers' own self-efficacy teaching beliefs as well as their confidence in using UDL to teach computing to students with disabilities increased significantly. The results suggest that exposure to CT learning experiences can have a positive impact on pre-service teachers' perceptions of the importance of CT/CS for their students and increase their self-efficacy in teaching it. These findings are important given that prior work with pre-service teachers has primarily focused on general education study while our study shows that special education pre-service teachers also see the importance of bringing computational learning experiences to students with disabilities. Our results are consistent with prior work that has shown that general education pre-service teachers see CT/CS as important for K-12 students [17][10][4].

Given the importance of teacher self-efficacy beliefs in influencing their classroom practice and impact on student outcomes [6], our findings are important for pre-service teachers to bring computing experiences into their future classrooms. If pre-service teachers see the value of CT for students with disabilities and feel confident in integrating CT, it is likely that they will bring computational learning experiences to their classrooms. Future research in this area should examine whether and how pre-service teachers translate their learning about CT and lessons on CT integration into actual classroom practice. One potential way to study this would be to follow pre-service teachers into their student teaching experience and implement CT integrated lessons that they develop as a part of their teacher education coursework.

Our findings from the case study scenario shows that pre-service teachers seem to use certain UDL strategies over others. In particular, multiple means of engagement had the highest number of codes, which is not surprising given that it is similar to making

accommodations for students with disabilities and something special education programs focus on. In addition, multiple means of representation was brought up the least. This could be something because it would require pre-service teachers to change what was in the lesson considerably. Without a complete understanding of CT/CS tools, it might be hard for them to come up with ideas on how to provide multiple means of representation. These findings are similar to how in-service teachers emphasize some UDL principles over others [7]. Israel and colleagues found that in-service teachers multiple means of action and expression was emphasized to a lesser extent than multiple means of engagement or multiple means of representation [7] whereas in this study multiple means of representation was least emphasized.

This finding that pre-service teachers can apply UDL strategies to support students with disabilities is promising and needs to be further explored. As Israel and colleagues have recommended, the three UDL principles address different student needs and "were not intended to hold equal weight as they are context- and student-dependent" [7]. Thus, future work should examine whether and how pre-service teachers use UDL principles to support different needs of the students and kinds of support they need to implement UDL strategies into their own lessons.

CT and CS are increasingly integrated into K-12 education for students. Yet, insufficient support exists to help teachers support and engage students with disabilities in accessing and experiencing success with CT and CS principles and activities. First step in increasing access to computing for students with disabilities is to educate future teachers on how to integrate computational thinking into their instruction. Our results suggest that integrating CT into special education teacher preparation coursework can have a positive impact on how pre-service teachers see the value of bringing computational practices to students with disabilities. Future work in this area should focus on whether and how teacher candidates bring computational thinking into their classroom instruction and/or different pacing of the curriculum.

ACKNOWLEDGMENTS

The authors thank the National Science Foundation for support under award 1936440. All opinions reflected in this paper are those of the authors and not necessarily those of the National Science Foundation. We also thank the many faculty who participated in this study and helped us to collect data from their students.

Appendix A : Case Study Scenario

Context: At the beginning of the school year, the elementary and special education teachers with whom they work attended a workshop on integrating computational thinking (CT) and computer science (CS) into their instruction. In this workshop, you focused on how to bring concepts such as decomposition, sequencing, and debugging into instruction.

Your new co-teacher, Ms. Martin, mentioned that she has been attempting to bring CT into her elementary classroom for the past two years. She confided in you that although she is excited to continue to explore ways to teach CT this year, several students struggled and seemed disengaged. Knowing that you will be working closely together this year, she suggested that you co-plan and co-teach the lessons. . You both agree that taking a Universal Design for

Learning (UDL) based approach will be key to successfully teach CS and CT this year!

Your co-taught classroom has a diverse group of learners that includes students who are bilingual, those who receive Gifted and Talented services, and students with disabilities who have individualized education programs (IEPs).

(1) Lauren has a specific learning disability in reading. She often does not ask for help because she does not want others to know she struggles. Lauren tended to work for a bit, but then begins sketching or doing something else that is off-task;

(2) Connie has autism and although she has good communication skills, she often struggles during collaborative activities. There are a couple of students in the class that Connie knows well and likes to work with. On many occasions, though, Connie becomes frustrated and asked these friends to do her work for her.

(3) Robert has limited mobility and attention issues due to a traumatic brain injury. He uses an electric wheelchair. He requires assistive technology including typing instead of handwriting. He also uses a modified stylus with a larger grip when using touch-screen devices.

You and Ms. Martin decide to meet during your co-planning periods to brainstorm some ideas about how to apply UDL to the Scratch shapes activity

The next step you take is to look at the UDL framework and the UDL in CS table to brainstorm ways in which the three UDL principles can be applied within this lesson. For each UDL principle, find at least one UDL feature that you notice and one feature you would add. (Re-read the context if needed)

- (1) Multiple Means of Engagement (The “WHY” of learning): What UDL feature(s) did you notice in the lesson?
- (2) Multiple Means of Engagement (The “WHY” of learning): What UDL feature(s) could be added to the lesson?
- (3) Multiple Means of Representation (The “WHAT” of learning): What UDL feature(s) did you notice in the lesson?
- (4) Multiple Means of Representation (The “WHAT” of learning): What UDL feature(s) could be added to the lesson?
- (5) Multiple Means of Action Expression (The “HOW” of learning): What UDL feature(s) did you notice in the lesson?
- (6) Multiple Means of Action Expression (The “HOW” of learning): What UDL feature(s) could be added to the lesson?

REFERENCES

- [1] [n.d.]. About Universal Design for Learning, howpublished = <https://www.cast.org/impact/universal-design-for-learning-udl>, note = Accessed: 2021-08-11.
- [2] Emily C. Bouck, Phil Sands, Holly Long, and Aman Yadav. 2021. Preparing Special Education Preservice Teachers to Teach Computational Thinking and Computer Science in Mathematics. *Teacher Education and Special Education* (2021). <https://doi.org/10.1177/0888406421992376>
- [3] Elisa Nadire Caeli and Aman Yadav. 2020. Unplugged Approaches to Computational Thinking: a Historical Perspective. *TechTrends* (2020). <https://doi.org/10.1007/s11528-019-00410-5>
- [4] Yu-hui Chang and Lana Peterson. 2018. Pre-service Teachers' Perceptions of Computational Thinking. *Journal of Technology and Teacher Education* (2018).
- [5] National Research Council. 2010. *Report of a Workshop on the Scope and Nature of Computational Thinking*. Technical Report. Washington, DC.
- [6] Sherri Gibson and Myron H. Dembo. 1984. Teacher efficacy: A construct validation. *Journal of Educational Psychology* (1984). <https://doi.org/10.1037/0022-0663.76.4.569>
- [7] Maya Israel, Gakyung Jeong, Meg Ray, and Todd Lash. 2020. Teaching elementary computer science through universal design for learning. In *SIGCSE*. <https://doi.org/10.1145/3328778.3366823>
- [8] Maya Israel, Jamie N. Pearson, Tanya Tapia, Quentin M. Werfel, and George Reese. 2015. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers and Education* (2015). <https://doi.org/10.1016/j.compedu.2014.11.022>
- [9] Gary Lewandowski, Dennis J. Bouvier, Robert McCartney, Kate Sanders, and Beth Simon. 2007. Commonsense computing (episode 3): Concurrency and concert tickets. In *Third International Computing Education Research Workshop, ICER'07*. <https://doi.org/10.1145/1288580.1288598>
- [10] Chrystalla Mouza, Hui Yang, Yi Cheng Pan, Sule Yilmaz Ozden, and Lori Pollock. 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* (2017). <https://doi.org/10.14742/ajet.3521>
- [11] U.S. Department of Education. 2011. *Schools and staffing survey (SASS)*. Technical Report.
- [12] Digital Promise. 2018. *Computational Thinking for a Computational World*. Technical Report.
- [13] Meg J. Ray, Maya Israel, Chung Eun Lee, and Virginie Do. 2018. A cross-Case analysis of instructional strategies to support participation of K-8 students with disabilities in CS for all. In *SIGCSE 2018 - Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. <https://doi.org/10.1145/3159450.3159482>
- [14] Kathryn M. Rich, Aman Yadav, and Christina V Schwarz. 2019. Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education* (2019).
- [15] Peter J. Rich, Ross A. Larsen, and Stacie L. Mason. 2021. Measuring teacher beliefs about coding and computational thinking. *Journal of Research on Technology in Education* (2021). <https://doi.org/10.1080/15391523.2020.1771232>
- [16] Jeannette M. Wing. 2006. Wing, J. M. (2006). Computational thinking. *Commun. ACM* (2006).
- [17] Aman Yadav, Chris Mayfield, Ninger Zhou, Susanne Hambrusch, and John T. Korb. 2014. Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education* (2014). <https://doi.org/10.1145/2576872>