2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #50756

Work-In-Progress: Designing Curricular Materials to Support Teacher Learning for Interdisciplinary Computer Science Education

Dr. Shannon Gooden Davidson, The University of Alabama
Hazel Altindis, The University of Alabama

Dr Altindis is an Assistant Professor of Mathematics Education at the Department of Curriculum and Instruction at the University of Alabama.

Designing Curricular Materials to Support Teacher Learning for Interdisciplinary Computer Science Education (Work in Progress)

Computer science education has gained widespread attention in K-12 schools in recent decades due to its increasing importance to the 'heart of our economy and the way we live our lives' [1, 2]. Given the computing-intensive nature of our society and the skill sets needed for $21^{st\text{-}c}$entury careers, including computer science learning in K-12 classrooms, this is imperative. To this end, it is critical that K-12 teachers feel prepared to teach computer science concepts and practices as part of their curricula and pedagogical repertoire. Yet even as standards have been adopted and initiatives advocating for more widespread computer science education have been developed at national and state levels across the U.S. [1, 3], many teachers feel unprepared to teach computer science if they have not had opportunities to become confident and competent in learning the pedagogical and content knowledge related to computer science [4]. Additionally, given the constraints of time, resources, and competing priorities in the K-12 classroom, it is important for teachers to have an understanding of how to meaningfully integrate computer science concepts and practices with other disciplinary learning by designing and implementing activities that take an interdisciplinary approach [5].

While professional development has been studied as an effective way to prepare teachers for computer science education instruction, much of this research has focused on supporting in-service teachers specifically teaching computer science at the secondary level [3, 5-6]. As such, teacher candidates are often left with limited curricular training to help them manage the challenge of learning about computer science concepts and integrating computer science into their instruction in meaningful ways [3, 6-7]. This is particularly true for elementary pre-service teachers who may see themselves as having less expertise or self-efficacy related to computer science and other disciplines related to science, technology, engineering, and mathematics (STEM) than their secondary counterparts [7-9]. In contrast, this approach embeds interdisciplinary computer science learning within required pre-service coursework and positions curricular materials themselves as educative tools for teacher learning. Rather than focusing exclusively on content acquisition, the modules are intentionally structured to develop pedagogical design capacity and interdisciplinary noticing. By treating curricular materials as vehicles for teacher learning—not merely classroom implementation—this work extends conversations about educative curriculum and teacher preparation into the domain of computer science integration.

Taking these concerns into account, this Work-in-Progress describes our ongoing efforts to develop and pilot an interdisciplinary computer science mini-curriculum designed to support pre-service teacher learning about computer science concepts, pedagogy, and integration with science and mathematics instruction. The goal of this work is to support pre-service teacher learning about computer science through an integrated approach by linking computer science concepts with other STEM-related disciplinary learning. We anticipate that teachers will become better equipped to design and implement lessons that highlight computer science concepts as central to and embedded within other subject areas. In the following sections, we define and operationalize computer science education within the context of this work; describe the interdisciplinary computer science curricular materials and their purpose; outline our plan for collecting and analyzing data related to this project; and share preliminary expectations for outcomes and contributions related to this work.

Computer Science Education in K-12 Contexts

According to the *K-12 Computer Science Framework* [1], computer science is defined as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their implementation and their impact on society" [10, p.6]. From a disciplinary standpoint, computer science education is different from, but related to, aspects of computer and technology education that are often already included in K-12 education such as computer literacy, educational technology use, digital citizenship, and information technology [1, 10]. Instead of focusing simply on computer use and application, computer science education efforts aim to support students to know why and how computers work to provide a basis for a deep understanding of computer use and the relevant rights, responsibilities, and applications [1].

A primary aspect of computer science education is that of computational thinking. Computational thinking is the ability to solve problems at multiple levels of abstraction to design solutions, create new tools, and communicate innovations across manifold computer-based and non-computer-based systems [1, 11-12]. It includes concepts such as algorithmic thinking, decomposition, repetition and pattern recognition, debugging, automation, and sequencing. While these skills are foundational to understanding the how and why of computing and computer programming, the concepts related to computational thinking are universally applicable and critically important across all disciplines. When integrated with other subjects, computer science education increases students' motivation, participation, and interest across subject matter domains [13]. Additionally, through the explicit linking of concepts and practices related to computational thinking with other disciplinary concepts and practices, student learning is bolstered in terms of problem-solving, creativity, and critical thinking [11-14].

Because of the broad applicability and transferability of computational thinking skills, K-12 teachers may have familiarity with computer science concepts through their teaching of other subject areas even if they are not explicitly aware of their knowledge. As an example, the practice of engaging in systematic error analysis in mathematics to determine where a mistake in computation or operationalization has taken place could be thought of as analogous to the process of debugging to systematically analyze lines of code to identify and localize a problem in program performance. Similarly, concepts such as sequencing, recursion, and repetition in programming and computational thinking mirror concepts found in biological science such as cellular division and replication, stages of life, and metabolic processes like the Krebs cycle.

As intuitively pleasing as these overlaps between computer science constructs and other disciplinary concepts might be, K-12 teachers may not make such links for students in the classroom without explicit consideration of them for themselves [2, 5]. The curricular materials described in the next section aim to support pre-service teachers to take up these connections for themselves by drawing their attention to cross-cutting connections related to computer science and computational thinking. In this way, the educative materials aim to engage pre-service teachers with computer science concepts in ways that support the development of their content knowledge and pedagogical knowledge related to computer science teaching and learning, and bridge key aspects of computational thinking with content and practices related to K-12 science and mathematics.

Development of Curricular Materials

This work takes an iterative design-based approach [15] to explore pre-service teacher learning of content and pedagogical knowledge related to integrated approaches to computer science after interacting with the interdisciplinary computer science educative materials developed at [a large southeastern university]. Our research team—comprised of four teacher educators with expertise in science, mathematics, computer science, and pre-service teacher education—has developed three distinct computer-based modules designed to support pre-service teacher candidates to learn fundamental concepts related to computer science; to connect these concepts to other focal disciplines (i.e., science and mathematics) from an integrated perspective; and gain knowledge and skills in designing and implementing computer science-related lessons and activities through an interdisciplinary approach.

Collectively, the three modules have five primary objectives in mind to provide pre-service teachers with opportunities to engage in meaningful learning related to computer science through an integrated and interdisciplinary approach. These include supporting pre-service teachers to:

1. Develop knowledge of and familiarity with K-12 computer science concepts and standards
2. Recognize the intersections of computer science concepts and the disciplinary content and practices of science and mathematics
3. Engage with high-quality instructional strategies for taking an integrated approach to teaching computer science concepts as interconnected with disciplinary content and practices
4. Explore innovative tools, platforms, and resources for teaching computer science concepts at the appropriate developmental and grade levels
5. Engage in collaborative reflection with peers and teacher educators around integrated computer science teaching and learning through an interdisciplinary approach

Below are brief descriptions of each module and its related activities. Table 1outlines the focus of each module.

Table 1. Summary of Modules Learning Goals

	Module 1	Module 2	Module 3
Focus	Understand CS concepts & standards	Identify interdisciplinary connections	Design for teaching and learning

Module 1: Exploring Computer Science Concepts and Standards

This module introduces basic computer science concepts related specifically to computational thinking such as sequencing, repetition, debugging, conditionals, and decomposition. Pre-service teachers are provided opportunities to engage with these concepts through play on coding platforms such as Code.org, Scratch, and using Bee Bots, along with unplugged activities. Then, the module introduces the K-12 Computer Science Standards [1] alongside the Next Generation Science Standards and the Common Core Mathematics which are typically more familiar to pre-service teachers due to their science and math foci. Pre-service teachers are then asked to first choose one grade-specific computer science standard or concept of interest and create a graphic organizer, table, or concept map to conceptualize interdisciplinary overlaps between their computer science standard and the other educational standards they have explored. For example, a pre-service teacher focused on conditional logic (if-then scenarios) as a concept for grade 8 might link that same concept with x/y linear functions in mathematics and cause and effect as a cross-cutting concept in science. They are then tasked with coming up with as many cross-disciplinary examples as possible between computer science, math, and science.

Module 2: Linking and Integrating Disciplinary Content and Practices

This module builds on the previous to support pre-service teachers to explore a variety of teaching resources to help them design an activity to integrate a focal computer science concept into a lesson related to another core content area (i.e., science or mathematics) for a specific grade level. First, pre-service teachers are tasked with exploring and analyzing examples of lesson plans and curricula that integrate computer science with science and/or mathematics. From there, pre-service teachers compare disciplinary practices across computer science, science, and mathematics and develop a graphic organizer as a 'disciplinary crosswalk' in an effort to draw explicit attention to the overlaps. Pre-service teachers are then tasked with drafting a student-centered activity that integrates at least one grade-appropriate computer science concept or practice into a science- or math-related inquiry-based learning activity. This draft activity is then shared with peers and teacher educators for feedback and discussion. See Table 2 for an example of a disciplinary crosswalk for K-2.

Table 2. Interdisciplinary Alignment of K–2 Computer Science Standards

K-12 CS Standard code	Grade	CS Standard	Math connection	Literacy connection	Science/Social science connection
1A-CS-03	K-2	Describe basic hardware and software problems using accurate terminology	Testing and refining computational artifacts MP1: Make sense of problems and persevere in solving them	Communicating computing problems W.3.2: Write informative texts to examine a topic and convey ideas clearly	Describing hardware/software problems K-PS2-1: Plan and conduct investigations

Module 3: Designing for Interdisciplinary Learning

This module once again builds on the previous two to support pre-service teachers to expand on their initial activity idea by developing it into a full lesson plan that takes an integrated approach across computer science and other disciplines. In this module, pre-service teachers first watch videos of computer science integrated lessons from practicing in-service teachers in order to see real world examples in action beyond written lesson plans. They are then tasked with returning to their draft activity from Module 2 and carefully consider peer and instructor feedback. Drawing on this feedback, the pre-service teachers are asked to build out their activity into a more fully formed lesson by including learning intentions/objectives, opportunities for student sense-making and discussion at both small and whole group levels, and the development of an assessment for learning that includes particular attention to the computer science connections in place. As the capstone of the modules, pre-service teachers are then asked to use a technology or platform of choice (e.g., Scratch, Canva, or a short video) to create a short (10 minutes or less) narrative presentation describing their lesson development and their learning across the modules.

Piloting the Modules

The modules shared above were designed by our research team over the course of the spring and summer of 2025 and so they have not yet been piloted with pre-service teachers. Initially, these modules were intended to be implemented in the context of an educational technology course (‘TECH200’) for undergraduates. While TECH200 is a prerequisite course for all K-12 teacher candidates, it is also an elective course that can be taken by any undergraduate at the university. Therefore, when we first hoped to pilot the modules in the fall of 2025, there were little to no teacher candidates enrolled in the course during the fall semester.

To mitigate this issue and to focus more pointedly on the interdisciplinary intentions of the modules, we are currently piloting these modules Spring 2026 across two pre-service education courses: SCI304-Elementary Science Methods and STEM300-Early Childhood STEM Approaches. In the Spring of 2026, approximately 100 elementary pre-service teachers are enrolled across all sections of these courses (two sections of SCI304 and three sections of STEM300) and will pilot the three modules as part of the regular coursework aligned with the goals and proficiencies of the courses. While all pre-service teachers enrolled in these courses will complete the modules, our team will collect data from those who volunteer to participate in the research aspects of this study. The research questions guiding this work are:

1. How do teacher candidates reflect on and describe their learning of content and pedagogical knowledge related to integrated approaches to computer science after engaging with the interdisciplinary computer science modules?
2. How do the interdisciplinary computer science modules influence teacher candidate attitudes and self-efficacy toward integrating computer science concepts with other disciplinary subject learning in their teaching practice?

Those pre-service teachers self-selecting to participate in the research components will complete a pre- and post-survey to assess their understandings of the targeted computer science concepts, their familiarity with the K-12 computer science standards, their self-efficacy toward teaching computer science broadly, and their self-efficacy toward teaching computer science through an integrated interdisciplinary approach. They will also participate in a post-course focus group interview to more deeply examine the impact of the modules on elementary pre-service teacher learning.

Data Analysis

We will use a mixed-methods approach, and pre- and post-survey data will be analyzed using descriptive statistics and paired-samples tests to examine shifts in understanding of computer science, familiarity with K–12 standards, and self-efficacy for interdisciplinary CS integration. Internal consistency of survey scales will be assessed (e.g., Cronbach's alpha), and effect sizes will be calculated to estimate the magnitude of change. Focus group transcripts will be analyzed using thematic analysis, combining deductive coding based on the research questions (e.g., pedagogical knowledge, interdisciplinary integration, self-efficacy) with inductive coding to identify emergent themes. Findings will be triangulated across data sources to strengthen validity and provide a comprehensive account of teacher candidate learning.

Implications

While this work is ongoing, we anticipate having preliminary findings to share as part of our presentation at the ASEE 2026 conference. Beyond the immediate pilot context, this work contributes a set of design principles for interdisciplinary computer science education in teacher preparation. Specifically, the modules foreground (a) explicit cross-disciplinary mapping of standards and practices, (b) structured opportunities for design and reflection, and (c) scaffolded progression from conceptual exploration to lesson enactment. These principles are not bound to elementary science methods courses alone but may inform broader efforts in interdisciplinary STEM and engineering education contexts where computational thinking is positioned as a cross-cutting practice rather than a standalone subject.

In engineering education contexts, where systems thinking, modeling, and iterative design are central, the integration of computational thinking across disciplinary boundaries is particularly salient. The crosswalk and lesson-design components of our modules may serve as a template for helping pre-service teachers conceptualize computing not as an add-on tool, but as epistemically aligned with engineering practices such as problem decomposition, iterative refinement, and debugging. In this way, the work contributes to broader conversations about preparing teachers to support K-12 engineering literacy through computationally rich, interdisciplinary instruction.

References

1. Computer Science Teachers Association (CSTA). (2017). Computer science standards. Computer Science Teachers Association. Retrieved September 1, 2022, from https://www.csteachers.org/page/about-csta-s-k-2
2. Luo, F., Ijeluola, S. A., Westerlund, J., Walker, A., Denham, A., Walker, J., & Young, C. (2023). Supporting Elementary Teachers' Technological, Pedagogical, and Content Knowledge in Computational Thinking Integration. *Journal of Science Education and Technology, 32*(4), 583-596.
3. Gray, J., Haynie, K., Packman, S., Boehm, M., Crawford, C., & Muralidhar, D. (2015, February). A mid-project report on a statewide professional development model for CS principles. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 380-385).
4. Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education, 26*(4), 235-254.
5. Leyzberg, D., & Moretti, C. (2017, March). Teaching CS to CS teachers: Addressing the need for advanced content in K-12 professional development. In *Proceedings of the 2017 ACM SIGCSE technical symposium on Computer Science Education* (pp. 369-374).
6. Milliken, A., Cody, C., Catete, V., & Barnes, T. (2019, July). Effective computer science teacher professional development: Beauty and joy of computing 2018. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 271-277).
7. Angeli, C., & Valanides, N. (2005). Preservice elementary teachers as information and communication technology designers: An instructional systems design model based on an expanded view of pedagogical content knowledge. *Journal of Computer Assisted Learning, 21*(4), 292–302.
8. Banilower, E. R., Smith, P. S., Malzahn, K. A., Plumley, C. L., Gordon, E. M., & Hayes, M. L. (2018). Report of the 2018 NSSME+. *Horizon Research, Inc.*
9. Shernoff, D. J., Sinha, S., Bressler, D. M., & Ginsburg, L. (2017). Assessing teacher education and professional development needs for the implementation of integrated approaches to STEM education. *International Journal of STEM Education*, *4*(1), 13–13.

10. Tucker et al. (2003). A model curriculum for K-12 computer science: Final report of the ACM K-12 task force curriculum committee. Retrieved from http://www.csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2nd-Ed.pdf
11. Lee, S. J., Francom, G. M., & Nuatomue, J. (2022). Computer science education and K-12 students' computational thinking: A systematic review. *International Journal of Educational Research*, *114*, 102008.
12. Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.
13. Sáez-López, J. M., Sevillano-García, M. L., & Vazquez-Cano, E. (2019). The effect of programming on primary school students' mathematical and scientific understanding: educational use of mBot. *Educational Technology Research and Development*, *67*(6), 1405-1425.
14. Witherspoon, E. B., Schunn, C. D., Higashi, R. M., & Shoop, R. (2018). Attending to structural programming features predicts differences in learning and motivation. *Journal of Computer Assisted Learning*, *34*(2), 115-128.
15. Saldana, J. (2015). *The coding manual for qualitative researchers.* Sage Publications.