How Preservice Teachers Simplify Computational Thinking Concepts During Elementary Science Lessons: Strategies and Challenges

Subject/Problem

Computational thinking (CT) is increasingly recognized as a foundational skill for K-12 students, and elementary preservice teachers play a crucial role in integrating CT into science curricula. Defined as a problem-solving process including decomposition, pattern recognition, abstraction, and algorithmic thinking (Wing, 2006), CT has become essential for contemporary education as educational systems mandate its integration into curriculum standards. Preparing preservice teachers (PSTs) to effectively implement CT instruction has emerged as a crucial research area, particularly within elementary science education where dual learning objectives create unique pedagogical challenges.

Background Literature

Recent scholarship has established theoretical foundations for CT integration in teacher education. Ragonis et al. (2024) propose the Four Pedagogies for Developing Computational Thinking framework, advocating implementation through active learning, project-based learning, product-based learning, and context-based learning across all subject areas. This represents a paradigmatic shift from viewing CT as exclusively computer science toward integrated, cross-curricular approaches. Waterman et al. (2019) propose three integration levels: identifying existing connections, enhancing current connections, and extending units with explicit CT activities.

Empirical research demonstrates the effectiveness of hands-on, technology-mediated approaches in developing PSTs' CT competencies. Tank et al. (2025) examined 27 preservice elementary teachers using Scratch and ScratchJr, revealing that hands-on activities significantly enhanced CT understanding and pedagogical skills. Similarly, robotics platforms have proven effective, with Jaipal-Jamani and Angeli (2016) finding statistically significant improvements in science content knowledge, self-efficacy, and algorithmic thinking among 21 elementary PSTs.

Despite documented successes, research reveals significant challenges that must be addressed for sustainable CT implementation. Butler and Leahy (2021) worked with 51 PSTs and demonstrated that developing deep CT understanding and sophisticated PCK required sustained engagement over two years. Brief interventions may yield limited outcomes and fail to provide sufficient foundation for meaningful classroom integration (Yadav et al., 2014).

A persistent challenge involves translating theoretical CT knowledge and technological tools into effective classroom practice. Tank et al. (2025) identified this theory-practice divide as fundamental, emphasizing that structured field experiences with authentic teaching opportunities are essential for developing PSTs' confidence and adaptability. Zha et al. (2019) found that integrating CT into coursework could support PSTs understanding of CT concepts, but they would require more support to actually apply their learning into practice.

Research has identified multiple strategies for addressing implementation challenges. Structured professional development frameworks prove critical, with Killen et al. (2023) employing design-based research over five years to implement dual-track approaches combining in-class CT instruction with collaborative, multi-month professional development opportunities. Research-practice partnerships provide another promising framework, as Dektor and Severance

(2023) found that collaborative approaches helped PSTs develop working definitions of CT and enhanced their efficacy for coherent instruction.

Addressing misconceptions requires targeted interventions. Sadik et al. (2017) found that while PSTs could provide basic CT definitions, some held misconceptions such as equating CT with algorithm design or suggesting trial-and-error approaches. Collaborative learning models show promise, with Ketelhut et al. (2019) finding that PSTs working with mentor teachers perceived CT integration as more feasible in their teaching contexts.

Research Gaps

Despite documented successes in computational thinking integration, significant gaps remain regarding how preservice teachers translate abstract CT concepts into accessible instruction for elementary students within authentic science contexts. Most studies focus on PSTs' learning outcomes rather than examining the specific pedagogical moves, simplification strategies, and real-time adaptations they employ during live science instruction. Additionally, insufficient exploration exists regarding the practical instructional strategies PSTs develop to manage technical difficulties, resource constraints, and classroom management challenges that arise when implementing robotics and coding technologies in elementary science lessons. Current research emphasizes successful outcomes while overlooking the moment-to-moment decisions and adaptive strategies PSTs use to navigate simultaneous science content delivery and CT skill development. This research addresses these gaps by examining:

1. How do PSTs effectively simplify computational thinking concepts for elementary students within science lessons?
2. What specific instructional strategies do PSTs employ to manage the practical challenges of integrating robotics and coding into live science lessons?

Design

Study Context

Sixty-six preservice teachers (PSTs) from four elementary education cohorts at a southern U.S. university participated in a program to enhance understanding of CT in elementary education. The PSTs received nine days of CT training across two semesters, encompassing CT concepts, robot coding, and pedagogical integration strategies for elementary English Language Arts, Mathematics, Social Studies, and Science. Training incorporated embodied cognition (Kopcha et al., 2020), use-modify-create methodology (Lee et al., 2011), and project-based learning approaches (Gutierrez et al., 2018) through direct instruction, hands-on practice, and reflective discussions. PSTs participated in sample lessons at lower and upper elementary levels following Hurley's (2001) sequential model, demonstrating integrated TPACK components. Each training day concluded with collaborative lesson idea generation, and semester endings featured small-group lesson plan development and peer teaching sessions.

During their final teaching practicum semester, PSTs implemented integrated CT lessons with faculty support for planning and assessment co-development. Lessons followed Hurley's (2001) teaching sequence and required CT instruction using robots, coding, or unplugged activities, with integrated practices emphasizing small-group problem-solving and ongoing facilitation. This design provided authentic contexts for examining PSTs' pedagogical decision-making and adaptation strategies during live CT-integrated science instruction.

Participants

All participants across the four cohorts designed lessons integrating CT with an elementary discipline, and this study focuses on the eleven PSTs who developed integrated science lessons. The PSTs designed plans that fit the grades they were teaching for their internship, and collaborated with the other participants who shared the same grade level to develop the lessons. The PSTs' adapted the collaborative frameworks as needed to meet individual classroom requirements. The participants included three PSTs working with 1st grade, four with 2nd grade, two with 3rd grade, and two with 4th grade. Ten participants were female and one was male; nine identified as White and two as Black.

Data Collection

PSTs submitted their shared lesson plans to the training faculty along with assessment tools. PSTs recorded their lessons and submitted them to the course management system after receiving the consent from students and their parents or guardians. These videos were transcribed and all transcripts deidentified students. PSTs also submitted after-lesson reflections by replying to the questions of "1. Describe how you integrated the computing into the subject lesson.", "2. How do you think the integration of computing in the subject areas benefit students?", and "3. What challenges did you encounter when planning and teaching this lesson?"

Analyses and Findings

Analysis

Researchers read through the lesson plans and reflections, followed by the transcripts of each teaching experience. They then watched the videos of each lesson to look for nonverbal evidence to support the teaching transcripts and took notes of these videos. All qualitative data was analyzed using

Data analysis followed systematic thematic analysis procedures to examine how PSTs simplify CT concepts and manage practical implementation challenges. The process began with thorough familiarization through multiple readings of lesson plans, assessment documents, PST reflections, and lesson transcripts. Initial open coding segmented data into meaningful units using both in-vivo codes reflecting participants' own words and descriptive labels capturing observed phenomena.

For the first research question examining how PSTs simplify CT concepts, codes included instances where PSTs used analogies, physical demonstrations, or relatable explanations. For the second research question investigating how PSTs manage practical challenges, codes captured problem-solving strategies, classroom management techniques, and adaptation responses. Focused coding then grouped similar codes into categories such as "Use of Analogies," "Physical Simulations," "Troubleshooting Strategies," and "Classroom Management Techniques."

Theme development synthesized categories into overarching themes addressing each research question. For simplifying CT concepts, themes included embodied learning approaches, relational analogies using familiar contexts, and iterative problem-solving strategies. For managing practical challenges, themes encompassed proactive planning with flexible adaptation, collaborative learning with differentiated support, and resource management with technical troubleshooting.

Trustworthiness was ensured through prolonged data engagement, triangulation across multiple data sources, rich contextual descriptions for transferability, thorough documentation of coding processes for dependability, and clear linkage between interpretations and raw data for confirmability.

Findings

This section presents the findings related to two key research questions: (1) How Preservice Teachers (PSTs) effectively simplify computational thinking (CT) concepts for elementary students within science lessons, and (2) What specific instructional strategies PSTs employ to manage the practical challenges of integrating robotics and coding into live science lessons.

Simplifying Computational Thinking Concepts

PSTs employed a variety of strategies to make abstract computational thinking concepts accessible and engaging for elementary students, often by linking them to tangible experiences and existing knowledge.

Relatable Analogies and Scenarios:

PSTs began by introducing CT as providing "explicit instructions" to robots, much like giving "step-by-step instructions" for familiar tasks, emphasizing that "robots can not think like humans" and understand code as a "different language". The physical coding blocks were consistently described and used as "puzzle piece[s]" or "building blocks," a concept familiar to young children. Robot controllers were analogized to "PlayStation or Xbox controller[s]," leveraging students' prior experiences with technology. Complex CT structures like conditional statements ("if, then, else") were simplified through kinesthetic activities, where students physically "act like robots" following commands such as, "If you have a sister, then put your hands on your hips". This direct physical engagement helped bridge the abstract nature of conditionals to a concrete understanding.

Concrete Representation of Commands and Variables:

To further simplify, PSTs utilized physical coding tiles or cards representing basic movements (forward, left, right, stop) or repeat actions, enabling students to construct sequences of commands manually before or during robot interaction. They connected numerical inputs directly to robot actions; for instance, students counted squares on a map to determine the exact number of "forward" steps a robot needed to take. Variables such as "revolutions per minute (RPM)" were explained in simple terms as "the speed of the engine" of the robot, demonstrating how adjusting these numbers directly impacted the robot's speed or movement characteristics. When exploring wave properties, PSTs showed students how changing numerical values in the code would alter the height (amplitude) and speed (frequency) of the waves produced by the robot, making abstract physics concepts observable.

Integration with Science Content:

CT concepts were seamlessly integrated into diverse science topics, allowing students to apply their coding skills to real-world scientific phenomena:

- Animal Care Simulation: Students used mTiny robots to simulate parental care by programming the robot (offspring) to navigate to essential items like food, water, shelter, and protection.

- Pollination Process: Robots modeled bees pollinating flowers, with students coding movements between designated flower spots on a map.
- Natural Selection and Camouflage: A mBot Neo robot acted as a "hawk" preying on paper "moths." Students initially observed the robot eating easily seen yellow moths, then colored their moths to "camouflage" with the background. This allowed them to "investigate the role of camouflage in the process of natural selection and trait inheritance". Students also graphed the number of moths eaten, connecting the robot's actions to data analysis.
- Wave Properties: Robots were used to demonstrate how varying code parameters affected wave amplitude and frequency, providing a tangible understanding of these concepts.
- Animal Habitats: Pre-kindergarten students directed mTiny robots to different habitat pictures on a map, associating animals with their correct environments.
- Disaster Recovery: Robots demonstrated their ability to navigate through simulated debris after a hurricane, with students predicting the robot's path based on an algorithm.
- Constellation Drawing: For older students, mBot Neo was programmed to draw constellations by plotting coordinates and measuring angles, connecting astronomy with geometry and coding.

Encouraging Exploration and Iteration:

PSTs fostered a growth mindset by allowing students "time to explore" with the robots and coding blocks. They normalized errors by emphasizing that "it's okay if it's not right the first time" and encouraged students to "try again" or "fix it", directly addressing the iterative nature of debugging in computational thinking.

Managing Practical Challenges

PSTs implemented diverse instructional strategies to navigate the practical challenges inherent in integrating robotics and coding into live science lessons.

Classroom Management and Setting Expectations:

A critical strategy involved establishing clear behavioral expectations from the outset. PSTs reminded students that "all of our classroom rules still apply" and that they needed to be "very careful with what I'm about to let you use". They managed student excitement by promising future "time to explore" after initial instruction or by gently prompting students to "slow down". To ensure equitable access to limited robots, PSTs used turn-taking strategies, such as "everyone can do one code and then switch off", and directly redirected off-task behavior with statements like "Sit down KJ" or "Stay over here. Focus".

Troubleshooting and Debugging:

PSTs actively facilitated troubleshooting. They modeled debugging themselves, remarking, "Oh, she only wants to taste we got to try again" when a robot didn't perform as expected. They encouraged student-led problem-solving by saying "let him fix it let him fix it" when students encountered errors. PSTs also provided direct, specific feedback on why robots might not be working, noting issues like "our moths were just a little off center" or clarifying that "your RP, your seconds Leonie is too long", thereby guiding students in adjusting code parameters (e.g., reducing movement seconds from 11 to 5 to prevent the robot from falling off the table). Technical glitches like robots not turning correctly ("it won't turn right") were addressed through direct intervention and guidance.

Differentiation and Support:

To cater to varied student needs, PSTs typically organized students into small groups (e.g., "groups of 4-6" or 2-3 per robot). This grouping allowed for more personalized "one-on-one instruction". They encouraged peer assistance, prompting students to "help her to if she needs help as a team help". For students who struggled, "pre-made code[s] to follow along with" or simplified instructions were provided. Conversely, advanced students were challenged to "create another pseudocode to a new direction". PSTs offered responsive support, intervening when students showed signs of frustration. They also ensured accessibility by reading directions aloud and allowing students to pull chairs closer to the rug if needed.

Logistical Management and Resource Needs:

PSTs managed materials by instructing students "do not fill it out until we get to that specific station". A recurring challenge highlighted by PSTs was the limited number of robots, with many emphasizing the need for "a class set of bots instead of just one" or "four or five, so that each group of students can participate at the same time". Adequate time and preparation were also cited as crucial, particularly for younger students who "had never seen a robot or had experience coding before". PSTs articulated a need for "someone who is knowledgeable on some of the robots" to "help troubleshoot" and for readily available "premade computing lessons already integrated into science, math, writing, social studies and reading". Technical issues encountered included color sensors not working correctly, robots not stopping without being fully turned off, and problems with recording devices like Swivl. Managing filming consent also posed a challenge, leading to "distractions" as PSTs focused on keeping non-consenting students out of view.

Overall, the integration of computing was seen as beneficial for enhancing engagement, problem-solving, and perseverance, despite the noted challenges related to resources, time, and technical support.

Contribution

This research addresses critical gaps in the computational thinking literature by moving beyond outcome-focused studies to examine the pedagogical mechanisms through which preservice teachers actually operationalize CT integration in elementary science contexts. While existing research extensively documents the effectiveness of CT integration initiatives (Tank et al., 2025; Tsai, 2023), limited attention has been given to the specific instructional moves, simplification strategies, and real-time adaptations that PSTs employ when making abstract computational concepts accessible to young learners during authentic science instruction.

Advancing Understanding of Pedagogical Translation Processes

Our findings reveal sophisticated pedagogical translation processes that PSTs develop to bridge the gap between complex CT concepts and elementary students' developmental capabilities. The identification of specific strategies—such as using kinesthetic activities to teach conditional statements, employing familiar analogies like "puzzle pieces" for coding blocks, and connecting numerical parameters directly to observable robot behaviors—provides concrete examples of how abstract computational concepts can be made tangible within science contexts.

These findings extend beyond the general pedagogical frameworks proposed by Ragonis et al. (2024) by documenting the moment-to-moment decisions PSTs make when simultaneously addressing science content objectives and CT skill development.

The seamless integration strategies observed across diverse science topics—from animal care simulation to wave properties demonstration—illustrate how PSTs create authentic connections between computational practices and scientific phenomena. This contributes to the limited literature on cross-curricular CT integration by providing specific examples of how computational tools can enhance rather than compete with science learning objectives, addressing Waterman et al.'s (2019) call for understanding different levels of CT integration in elementary contexts.

Informing Teacher Preparation and Professional Development

These findings have significant implications for teacher preparation programs struggling to prepare PSTs for the practical realities of CT-integrated science instruction. While Butler and Leahy (2021) emphasized the importance of sustained professional development, our research reveals the specific competencies PSTs need to develop: classroom management strategies for technology-rich environments, real-time troubleshooting skills, and differentiation approaches that accommodate diverse learners' needs with limited technological resources.

The identification of persistent challenges—such as managing limited robot access, addressing technical difficulties, and providing adequate preparation time—offers evidence-based insights for program designers. The PSTs' expressed need for "premade computing lessons already integrated into science, math, writing, social studies and reading" and "someone who is knowledgeable on some of the robots to help troubleshoot" provides specific guidance for institutional support structures that go beyond the general recommendations found in existing literature.

Addressing Equity and Accessibility Concerns

Our findings contribute to the growing literature on equity in CT education by documenting how PSTs navigate resource constraints while maintaining inclusive practices. The strategies observed—such as strategic grouping, peer assistance encouragement, and provision of "pre-made codes to follow along with" for struggling students—address Coenraad et al.'s (2020) concerns about CT implementation potentially upholding existing inequities. The fact that 88% of participating schools were Title I institutions provides valuable insights into CT integration in under-resourced contexts, an area identified by Ausiku and Matthee (2021) as critically understudied.

Enhancing Assessment and Evaluation Practices

While existing research has developed comprehensive assessment tools for measuring PSTs' CT competencies (Adler et al., 2022; Molina-Ayuso et al., 2022), our findings illuminate the assessment challenges PSTs face when evaluating student learning across both science content and computational thinking skills simultaneously. The observed integration of data analysis activities—such as graphing moth predation data in the natural selection simulation—demonstrates how PSTs create authentic assessment opportunities that capture dual learning objectives, addressing a significant gap in the current literature.

Implications for Sustainable Implementation

These findings provide essential insights for scaling CT integration initiatives beyond pilot programs. By documenting the actual resources, support systems, and preparation requirements necessary for successful implementation, this research contributes practical knowledge that can inform policy decisions and resource allocation in elementary schools. The detailed challenges and mitigation strategies observed offer realistic expectations and planning considerations for administrators and teacher educators working to implement sustainable CT-integrated science programs.

This research ultimately advances the field's understanding of how theoretical CT integration frameworks translate into classroom practice, providing the granular pedagogical insights necessary to support effective teacher preparation and successful program implementation in diverse elementary science contexts.