WIP: Challenging and Transforming Computing Education for Future K-12 Teachers

Elaine M. Hampton, Ph.D.
STEM Education Associates
Las Cruces, New Mexico
ElaineMHampton@gmail.com

Leilana Luna
Department of Computer Science
University of Texas at El Paso
El Paso, Texas
laluna@miners.utep.edu

Christopher Villa
Helix Solutions
El Paso, Texas
ccvilla@helixeval.com

Randy Taylor, Ph.D.
Helix Solutions
El Paso, Texas
rstaylor@helixeval.com

Elsa Villa, Ph.D.
Hopper Dean Center
University of Texas at El Paso
evilla@utep.edu

Mary K. Roy
Department of Computer Science
University of Texas at El Paso
El Paso, Texas
mkroy@utep.edu

***Abstract*— This innovative practice WIP paper describes an effort to introduce future teachers, who are primarily Hispanic and planning to be PK-8 teachers, to computational thinking for teacher preparation at a Hispanic Serving Institution located on the U.S.-Mexico border. An interdisciplinary team of education and computer science faculty is implementing efforts to increase pre-college student engagement in computational thinking (CT) along with increased equity, particularly for Hispanic and women. One aspect of these efforts is the modification of an existing CT course that will eventually be required for all teacher preparation students. The first pilot of this CT course was conducted in the fall 2023 and included quality pedagogical approaches to teach CT content to primarily Hispanic pre-service elementary teachers.**

Keywords*—recruitment and retention, computational thinking, teaching skills, preservice elementary teachers

I. Introduction

According to Code.org®, 90 percent of parents want their children to learn computer science (CS); yet only 57.2 percent of high schools offer CS courses [1]. This trend of few CS course offerings is likely to persist as indicated by a 2023 U.S. Department of Education Title II Report [2] stating that the discipline of computer science gained a mere 62 new teachers nationwide between 2019 and 2020 while physics gained 903; general science 5,382; and mathematics 9,148. Furthermore, Sara Sparks [3] with Education Week examined the need for more CS teachers and found a variety of approaches exist across the nation to train and certify CS teachers, such as the *WeTeachCS* program at The University of Texas-Austin, a program offering virtual and on-the-job professional development for Texas teachers to prepare them for challenging the state's computer science certification. In spite of such programs, ensuring all students receive quality computer science education from certified teachers remains a distant goal.

The University of Texas at El Paso (UTEP) has recently initiated multiple approaches to advance the quality and quantity of computer science education. One approach, which is the focus of this study, is to embed computer science and computational thinking (CT) into required coursework in UTEP's teacher preparation program. While the curriculum materials were not necessarily innovative, what is innovative is the implementation of a required course for teachers that applies CS concepts and tools. The students served are primarily Hispanic PK-8 preservice teachers in a predominantly low-income region of the U.S.

Complex negotiations regarding the constraints of tightly defined degree plans are being addressed, and the course pilot was implemented in Fall 2023. Three of the authors were on the teaching team for the course pilot; and three of the authors were evaluating the course via observations, interviews, and surveys. The following describes the pilot implementation and provides evidence of successes in the new course along with considerations for ways to improve the experience for students. The team also examined the course from the students' perceptions based on the Concerns Based Assessment Model (CBAM) [4], an approach to measure students' changes in concerns that may indicate advances in their identity as computer science teachers, especially those who are Hispanic, as the community in which this study took place is a majority Hispanic community. To the best of our knowledge, introducing computational thinking to primarily Hispanic preservice teachers is novel, which brings innovation to this study.

As a framework, this course drew from Wing's [5] understanding of CT and her call to reach pre-college audiences. "Computational thinking is a fundamental skill for everyone, not just for computer scientists… Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p. 34).

This course also addressed disparities in student outcomes in light of CAPE: Capacity for, Access to, Participation in, and Experiences of Equitable Computer Science Education [6] that posits "assessing equity not simply as an end product, but as an integral component to each element of the systems that support computing education" (p. 24). The course and its consideration of equity in preservice teachers' understanding of CT is a valuable addition to teacher education curriculum and has strong potential to impact thousands of K-12 Hispanic students across the El Paso border region through greater capacity, access, participation, and experiences in computer science education. Over 95 percent of the students in this course were Hispanic and were primarily interested in elementary education.

II. The Course Design

The course was designed to focus on aspects of CT such as decomposition, pattern recognition, abstraction, and algorithmic thinking. The instructors incorporated student-centered strategies with inquiry-based learning to solve problems while integrating concepts from art, English, math, physics, and history. Cooperative learning approaches [7] were foundational for all activities and strengthened students' participation through engaged interaction that moved them to think critically about problems posed with the use of technology tools.

The instructors found the first few class sessions troublesome. Students, who were primarily seeking early childhood education, seemed confused about how CT was different from computer science and why they were going to learn what they interpreted as "coding." Students appeared disinterested.

In one introductory activity, students created name tents, found a nearby partner, and introduced themselves. These connections among the students increased and deepened throughout the semester. As their comfort levels increased, students became more confident to speak up and participate. Throughout the semester, students worked cooperatively in structured groups. Even during individual assignments, students sat near each other and were allowed to pool knowledge in order to complete their work. This allowed for their resources to become "each other." While some students had more math skills, others were better at writing. As the majority of the students had strong Mexican roots, the majority of student-to-student interactions were in Spanish. All class communications were in English. They helped one another when help was needed.

The following describes the key lessons in the course that required manipulative technology learning tools. Since learning evolves from interaction with the tools and peers, efficient management of the tools and grouping was vital. Therefore, that management process is also described. The authors also provide ways that the learning experiences will be improved in the next iteration of the class in order to maximize preservice teachers' understanding of and ability to integrate CT into their instructional repertoire.

The first pilot of this CT class for education majors provided a wealth of information to incorporate into the second iteration of the class which will occur in Fall 2024. As the students' experiences working in groups proved successful, a guiding principle for development of the new course will be covering topics in such a way that students will work in groups to an even greater extent and take charge of their own learning process. The instructors' plan for the upcoming class will be to provide the preservice teachers with the challenge or problem, and then provide the appropriate support as they engage in the tasks to address these desired ways of learning listed below. These approaches are designed to help the students become more effective and confident problem-solvers.

- Design the project and the most effective development process.
- Assess what you have/need in the way of knowledge, skills, or resources.
- Divide the work among group members for mutual benefit.
- Locate resources (which could include asking peers or instructors).
- Produce and share artifacts that showcase lessons learned, for example, Miro online whiteboard, PowerPoint presentation, your final design product, or a program display.

A. Defining Computational Thinking (CT)

In the Fall 2023 pilot, students were given a puzzle displaying geometric triangles nested within larger triangles. They were asked to find the total number of all triangles. The triangles were complex enough that students could not simply count them. Students compared answers with their teammates, many of whom had applied successful strategies. These successful approaches were shared with the large class, highlighting their computational thinking. Following the triangle activity, students were given a copy of Wing's [5] relevant article describing CT along with a diagram of the four main components often used in CT. They then wrote a reflection on the ways their work with the triangles aligned with Wing's perspective on CT and volunteers shared ideas from their reflections. Students were actively engaged in these introductory activities and identified their strategy to solve the triangle puzzle as it related to CT.

To improve this introductory activity to deepen students' understanding of the concept of CT in the Fall 2024 class, the team will provide students with Wing's article first, and then ask them to generate their own ideas about what they think is meant by "computational thinking," what components could be included, and how they could apply CT to their own experiences. From their previous experience, the instructors have found that sharing important reflections on the lessons publicly on a platform, such as Miro, ensures participation (everyone in the class could see if you did your assignment!) and nudges the students toward more accurate and well-written responses. To strengthen this understanding, the team will include a group discussion of the question, "Is a computing device required for computational thinking? Why or why not?"

In the pilot, students were provided with a description of the general problem-solving process with an example of an application of the process to an everyday problem. Afterwards, students were asked to analyze a real-life problem of their choice, using the same process steps. To improve activities to help students understand problem solving skills in the Fall 2024 class, students will work in groups to discuss and generate their own ideas of a problem-solving process and compare/contrast that with the scientific method.

B. Binary Bracelets, Scratch, and Spheros

The binary bracelets lesson began with a short review of place value and groupings. While the students somewhat understood that binary uses only two digits, they expressed confusion about the representation of numbers in binary as well as translating from binary numbers to ASCII code. Each student received a small bag filled with different colored beads with precut strings and an instruction sheet. The instructors created an example bracelet translating each letter of a name to a decimal number and then to binary code. Then each student was tasked with creating their own name in binary beads. The instructors noticed some confusion about the instructions, but students did not ask the instructors for help. They asked their peers, and someone could usually answer. They shared materials as needed. While they created their bracelets, they focused on decomposition as they had to do multiple translations to reach the binary encoding of their name. Many began trying to translate their full name, moving straight to binary code. Eventually, every student translated each letter to decimal numbers, and finally binary code.

Scratch is a free online program that teachers can use with students to apply CT and problem solving to create their own digital story or animation. Scratch is useful for the early stages of creating a computing process and interacting with many different programming blocks with clear definitions, while not overwhelming them with options. In this pilot class, students were tasked with creating a story using Scratch that incorporated different elements such as broadcasts and loops.

Initially, students wrote a plan for their story; and when complete, they worked in small teams to add individual elements. These first stories were long, extremely detailed, and based on simple plots. As they began writing these stories in Scratch, they found that they were unable to create their story in Scratch at that level of detail. They worked together to find the most important points and simplify the story. They discovered the importance of abstraction.

The instructors provided a short lecture addressing concepts such as base code, pattern recognition, and loops which the students incorporated into their programs. The results were simple stories and very simple code. The instructors found that, in this instance, allowing students to work with little instruction led to creativity in their project, but the technical aspects were too basic — lines of code created were simple tasks. The instructors will modify the lesson so that the stories address multiple disciplines, and students will receive more instruction on the use of loops, patterns, and abstraction. Therefore, in the Fall 2024 iteration of this class, students will receive just-in-time examples of specific elements such as ways to make the character move across the screen. Rather than inventing a story plot, students will develop their Scratch story to teach a lesson from the desired content area.

Sphero is a technology learning tool that allows users to move the Sphero at different speeds and in different directions through block coding. The experiences described below indicate that the use of Spheros was effective in advancing the preservice teachers' CT understandings. In addition, there was evidence that embedded in the Sphero experiences were STEM concepts, teamwork, and collaboration.

Building on previous experiences, in this lesson, the students received more instructions—very clear and specific to the tasks—along with diagrams and images. Students continued working in groups and collaborating with peers to develop their designs to move the Sphero. The tasks were carefully organized from simple to complex, building upon new skills and experiences. The instructors monitored the activities and were available if help was needed. However, the instructors observed students relying on each other as they approached the tasks in multiple ways and seldom asked for help. The Sphero activities allowed students to learn using trial and error; tasks included speed and direction control as they programmed their Sphero to make 90-degree turns in the first lessons and eventually to maneuver complex mazes and designs that had been taped on the classroom floor. Finally, students created their own mazes with multiple turns and curves. One team had been struggling; and, when their Sphero successfully traveled the maze, the full class celebrated with that team's accomplishment.

There were challenges, and some students struggled and became frustrated. They began to withdraw from the activity and allowed others to take charge. The instructors intervened with gentle suggestions and hints. Managing the curves and angles seemed particularly challenging; so, the instructors implemented on-demand learning with a quick explanation of basic geometry and angles. At one point, the instructors observed the teams applying the physics of speed with no assistance -- managing the Sphero's speed as it encountered an angle or a curve. They appeared more confident and attempted more challenging shapes. One group challenged themselves to write their program so that their Sphero followed the outline of the university's logo, a miner's pick.

As the student teams quickly picked up the basic Sphero control, they were able to create their own activities. In the future classes, there will be fewer pre-defined Sphero activity cards and more room for students to create. The instructors will highlight the content learning that is inherent in the Sphero activities. For example, the students used Sphero as a battering object to knock over a structure made of small plastic items. The students incorporated repeated trials with increased speed. The lesson was learned, and it was time to apply the science wording: The acceleration of an object is directly proportional to the net force and indirectly proportional to the object's mass.

III. Evaluation

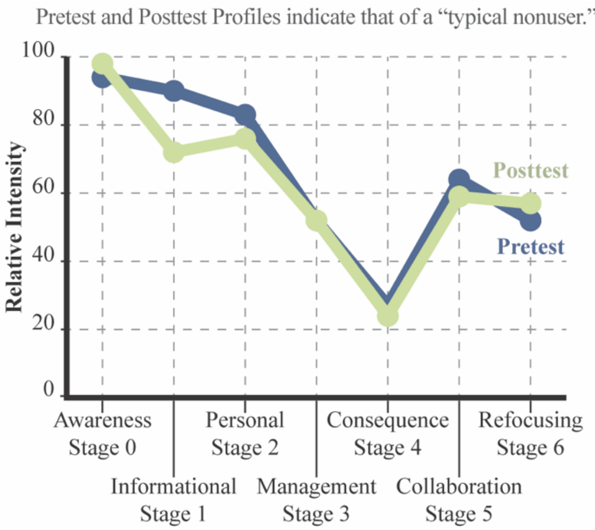
The project aimed to equip preservice teachers with tools for their teaching careers through exposure to CT. Evaluators assessed the participants' mindsets regarding these concepts (beyond just their satisfaction or academic achievement in the course) using the Concerns-Based Adoption Model (CBAM) [7]. Specifically, the evaluators assessed the Stages of Concern component through the Stages of Concern Questionnaire (SoCQ) [4]. SoCQ comprises 35 items assessing the nature of the concerns held by respondents regarding an innovation.

The instrument categorizes concerns across seven levels, which are aggregated into three major categories: self-concern (Stages 0–2), management-concern (Stage 3), and impact-concern (Stages 4–6). The preeminent concerns of an individual can vary from having little or no knowledge about an innovation (more focused on the individual themselves) to increased knowledge, having concerns about managing time and resources, and optimizing their practices [8].

The survey items were tailored to the current context—replacing terminology from "the innovation" to "the computational thinking principles." Utilizing pretest-posttest methodology, data were collected at the start of the semester and again late in the semester, with responses matched using a unique identifier. Pretest and posttest results were interpreted as outlined in the SoCQ manual [4]. Following the manual's procedures, the evaluators developed the percentile scores for the pretest and posttest data, as illustrated in Figure 1 below.

The SoCQ manual discusses several aspects of the profiles to consider, including the relative positions of scores in Stages 1, 2, and 6. First, the pretest findings indicate that the Stage 1 average percentile score is larger than the Stage 2 average scores, with this pattern being reversed at the posttest. The SoCQ manual suggests that a higher Stage 1 score than Stage 2 is "open to and interested in learning more about the innovation" (p.40). Alternatively, a Stage 2 score higher than a

Figure 1.



Source: George, Hall, and Stiegelbauer, 2008, p. 42

Stage 1 score can indicate that "the concerns about an innovation's effect on personal position or job security usually are greater than the desire to learn more about the innovation" (p.40). While the highest scores were in the Self-Concern category at both pretest and posttest, the SoCQ results can reveal subtle changes in participant concerns over time. For example, the SoCQ manual states when the Stage 6 score "tails off or down," it indicates that one "does not have ideas that would potentially compete with the innovation" (p.42). Alternatively, when the Stage 6 score "tails up," it indicates that one "has ideas that he or she sees as having more merit than the proposed innovation" (p.42), which offers opportunities to observe shifts in student concerns over time.

Upon examining the pretest and posttest curves, it is evident that the Stage 6 score "tailed down" at the pretest, while the posttest scores "tailed off," or flatter. While these findings may indicate some change, it does not alter the interpretation of the profile pattern. In other words, both results suggest that the respondents did not develop competing ideas during the semester. However, the results suggest that participants' self-oriented concerns shifted from an interest in learning about the CT principles to how they may fit in their functional roles.

This aspect of the evaluation did not indicate significant growth related to the participants' concerns about teaching CT. It is possible that the dosage (ten class meetings) was not sufficient to observe substantial changes. Also, some content and the collaborative approach to learning may have been new to the students, mostly freshmen and preservice teachers. This informs the project leadership about modifications that might improve evaluation and implementation in future iterations of the class.

The evaluators asked students to indicate their satisfaction with the course. Most students stated that they were overall satisfied with the course and would recommend it to their peers (75% in each case). However, less than half of respondents stated that the course improved their understanding of the value of computational thinking to a great or very great extent for their career (44%) and academic preparation (31%). Some students suggested grounding the curriculum with practical examples or case studies of applying CT to the work of a teacher. This finding aligns with the results of the SoCQ questionnaire in that students may not feel that they have integrated their CT knowledge gains. Further, they might not have a perspective on how to apply CT pedagogy in the practical execution of a teaching career. Providing students with practical case studies might provide them with meaningful connections, potentially changing their SoCQ profiles. The SoCQ can be a powerful assessment tool due to its ability to provide insight into both broad and subtle aspects of respondents' concerns by providing implementers an opportunity to gain nuanced insights.

IV. Summary

Computational thinking is important for everyone. Young people's choices for their careers are stronger when they have these thinking skills. Teaching CT to teachers is a worthwhile way to provide this form of capital to their students, especially students living in low-income areas, such as our study site of El Paso, Texas. However, quality educational programs for teaching CT to teachers are emerging slowly, especially among Hispanic preservice elementary teachers. We share our experiences and lessons learned as we attempted to implement the most impactful learning experiences in the restricted time frame of a one-semester class. We identify practices that proved successful along with those that need modification and describe those modifications. Identifying the impact of this multifaceted and often complex set of learning experiences for preservice teachers and evaluating that impact was more challenging than we expected.

References

[1] Code.org/promote. "Why computer science." Retrieved on April 20, 2024.

[2] U.S. Department of Education, Office of Postsecondary Education. Preparing and Credentialing the Nation's Teachers: The Secretary's 13th Report on the Teacher Workforce. (July 2023). Available at https://title2.ed.gov/Public/TitleIIReport23.pdf.

[3] S. Sparks, "States are calling for more computer science classes. Now they need the teachers," Education Week, October 5, 2023.

[4] A. A. George, G. E. Hall, & S. M. Stiegelbauer, Measuring Implementation in Schools: The Stages of Concern Questionnaire. SEDL: Austin, TX. 2006. ISBN: 0-9777208-0-2

[5] J. Wing, "Viewpoint: Computational Thinking," Communications of the ACM, vol. 49, no. 3, pp. 33-35, 2006.

[6] C. Fletcher and J. Warner, "CAPE: A Framework for Accessing Equity throughout the Computer Science Education Ecosystem." In Viewpoints Education Communications of the ACM, vol. 64, no. 2, pp. 23-25, (February 2021).

[7] G. E. Hall, The Concerns-Based Adoption Model: A Developmental Conceptualization of the Adoption Process within Educational Institutions. Retrieved from http://files.eric.ed.gov/fulltext/ED095126.pdf, 1974.

[8] A. Dubey. Evaluating the Concerns of the In-Service Students of Fiji National University. International Journal of Information and Communication Technology Education, 12(4), 37-50. Fiji National University, Lautoka, Fiji, 2016.