Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #48531

Computer Science Professional Development for Middle and High School Teachers: Insights from Three Cohorts

Jina Wilde, The University of Texas at San Antonio

Research assistant involved in the development and delivery of professional development modules in computer science and STEM, as part of an NSF-funded research project.

Michael Zawatski, The University of Texas at San Antonio

.

Darean Wilde, The University of Texas at San Antonio

Research assistant involved in developing and delivering professional development modules in computer science and STEM, as part of an NSF-funded research project.

Emiliano Beltran, The University of Texas at San Antonio

Student researcher interested in how the delivery of professional development can impact a teacher's ability to influence students to pursue STEM (and more specifically, CS-related) degrees.

Dr. Amanda S. Fernandez, The University of Texas at San Antonio

Amanda S. Fernandez an Assistant Professor of Computer Science at the University of Texas at San Antonio.

Dr. Timothy Yuen, The University of Texas at San Antonio

Timothy T. Yuen is the Associate Dean for Undergraduate Studies in the College of Sciences at the University of Texas at San Antonio.

Computer Science Professional Development for Middle and High School Teachers: Insights from Three Cohorts

Abstract

Computer Science for San Antonio (CS4SA) was a computer science (CS) professional development program designed for in-service middle and high school teachers—educators actively teaching. CS4SA aimed to prepare teachers with essential CS knowledge and skills while expanding CS opportunities for Latinx and other underrepresented minority populations within a large, urban school district in South Texas. An Institutional Review Board approved this research.

The program engaged teacher participants through culturally responsive pedagogy, integrated professional learning communities, and project-based learning strategies. Teachers appreciated the collaborative nature of these approaches, which deepened their understanding and strengthened their professional networks. Over a pilot program and three cohorts (2020–2024), participants attended a Summer Institute, monthly workshops, and completed online modules aligned with state CS teacher certification standards.

This paper examines the program's instructional design and shares insights from participants, including those who returned as peer mentors. It also outlines adaptations to address challenges such as COVID-19 disruptions and limited district support. Findings indicate that teacher participants valued their professional development experience and quickly applied their new skills in the classroom. Many teacher participants integrated CS into their math and science lessons. They introduced CS concepts in after-school clubs, supported by program resources that enabled projects ranging from Unity game development to robotics and Scratch programming. One teacher secured funding for additional robotics resources, while another invited a software developer to discuss app development, demonstrating the real-world applications of CS in various industries.

Despite challenges such as school closures due to declining enrollment, staffing reductions, and limited district support, educators found creative ways to engage students through robotics and coding projects. CS4SA helped build a community among participants, allowing them to exchange ideas and resources. While the program's impact on expanding computer science education within schools was more limited than anticipated, it played an essential role in supporting teachers as they integrated CS into their classrooms. These findings highlight the role of professional development in supporting teachers as they integrate CS into their schools and classrooms.

Keywords

Computer Science Teacher, Professional Development, CSforAll, K-12 STEM Education, Culturally Responsive Teaching, Project-Based Learning

Introduction

Computer Science for San Antonio (CS4SA) was a teacher professional development (PD) program designed to equip a diverse group of STEM educators in a large, urban, predominantly Latinx school district in South Texas with computer science (CS) knowledge. Over three cohorts, beginning in August 2020 and ending in July 2024, CS4SA prepared teachers from non-CS backgrounds to introduce CS experiences and instruction into their classrooms to increase Latinx participation in CS. The program presented computer science and pedagogical content aligned with the state's high school CS teacher standards. It also assisted teachers in navigating the challenges of implementing CS teaching in diverse classrooms.

To support these goals, CS4SA designed and developed a professional development program that included instructional materials, workshops, tutoring, coding camps, and campus visits, where teachers observed presentations about university CS degree programs, deepening their understanding of CS applications and career pathways. The instructional materials included books on computer programming [1] and culturally responsive pedagogy [2], [3], offering theoretical and practical frameworks to support teachers' learning. Teachers also had access to online instructional modules and participated in workshops provided online synchronously and/or in-person, depending on the cohort.

This paper outlines the program design and relevant literature, describes its implementation (including the Summer Institute, workshops, and online modules), and presents insights from the pilot and three cohorts of teachers. The PD overview examines the development of the modules and projects, while the insights highlight broader teacher outcomes and overall impact. The paper concludes with insights into the program's implications and potential for similar future PD efforts.

Literature-Informed PD Design

Teacher professional development should equip in-service teachers with innovative ways of teaching and lead to meaningful changes in practice [4], [5]. Recent literature [6], [7], [8], [9] highlights effective CS PD programs that augment content knowledge, broaden student participation, and promote a strong professional identity and positive attitude toward teaching. The goal was to create a collaborative network of CS educators across a school district, preparing them to provide more CS opportunities in their diverse classrooms.

The program built professional learning communities (PLCs) among teacher participants across cohorts while increasing their knowledge of and enthusiasm for CS. Dogan et al. [10] define PLCs as environments where "teachers commit to a common vision of improving student learning, work collaboratively to find solutions to problems of practice, and evaluate the success of their efforts to improve pedagogy based on student achievement…" Incorporating PLCs encourages collaboration and promotes a culture of continuous improvement. Through community-building and reflection, teachers strengthen their professional identities and teaching practices [11]. Teacher participants from previous cohorts also served as peer mentors for current participants, offering guidance and support in completing the program and navigating school and district systems to create more CS opportunities for students. This type of community-building was essential for developing a base of advocates who could establish CS pathways across the district aligned with the project's goals. An emphasis was placed on culturally responsive pedagogical approaches to teaching CS as the project's goal was to increase Latinx participation in computer science. Culturally responsive pedagogy was presented at the beginning of CS4SA and used as the foundation of how teacher participants would introduce CS to their largely Latinx student populations [2], [12]. As they learned the CS topics, the project team worked with the teacher participants on how they would use their students' linguistic and cultural backgrounds as a platform for teaching CS. Teacher participants examined ways in which CS was relevant to their students' home and community lives and how that could be infused into the CS classroom. The purpose was to make computer science meaningful to their diverse classrooms and to create spaces for their students to see themselves as computer scientists.

CS4SA was initially envisioned as an in-person professional development program. However, due to COVID-19, the program began with a set of online modules, including both asynchronous content and synchronous workshops and tutoring sessions. As in-person activities gradually resumed, CS4SA transitioned to a hybrid model, blending online and in-person components. Throughout all cohorts, the core CS content was delivered via asynchronous online modules. In-person or synchronous online meetings focused on reviewing CS concepts, completing programming and other computing exercises, and discussing CS pedagogy, particularly strategies for increasing Latinx and underrepresented minority participation in CS and other STEM fields. As they were relatively new to computer science, it was important that the online modules were designed to engage the teacher participants as active learners.

Since most of the modules focused on programming, they were naturally designed using project-based learning (PBL), where teacher participants applied their knowledge by creating programs and technology projects, such as robotics or microcontroller-based systems. Lam [13] describes PBL as an inquiry-based method that engages students with meaningful problems, enabling them to construct various artifacts during the learning process. This approach accelerates CS education by promoting active participation and social interaction [14], which stimulates students' interests and deepens their knowledge through context-specific projects. PBL also offers students opportunities to apply their knowledge in real-world contexts. Throughout the program, teacher participants engaged in PBL through interactive exercises, including a summer coding camp. They received additional support via discussion boards and tutoring sessions, encouraging further exploration and understanding of the material. By integrating PBL with professional learning communities, the program equipped teachers with practical approaches to enhance educational outcomes in their classrooms and engage their students in meaningful culturally responsive CS experiences.

PD Overview

The CS4SA program included a comprehensive set of activities designed to prepare teachers for the Texas CS certification exam, Texas Examinations of Educator Standards (TExES) 241, and equip them with practical tools for teaching computer science. These activities included discussions on pedagogy, content modules aligned with TExES 241 domains [15], projects with feedback to refine skills, quizzes, and videos to reinforce learning, and application practice through coding exercises, such as building educational apps, games, or tools like calculators. CS4SA featured a three-week Summer Institute, six monthly in-person workshops, and 14 online modules. Participants earned over 240 continuing professional education hours, were reimbursed for up to two certification test attempts, and received a stipend upon completion of all program components.

Module Content and Structure. CS4SA began with 14 online modules aligned with the TExES 241 content, beginning with an emphasis on culturally responsive teaching practices. The first module set the stage for the program by connecting CS concepts to students' lived experiences, using resources including *Culturally Responsive Teaching: Theory, Research, and Practice* [2]. These initial modules helped teachers integrate students' cultural backgrounds into CS lessons while laying the foundation for more technical content. As shown in Figure 1, the introductory page for this module outlined its objectives, connected to relevant TExES standards for high school computer science [15], and earned continuing professional education (CPE) hours.

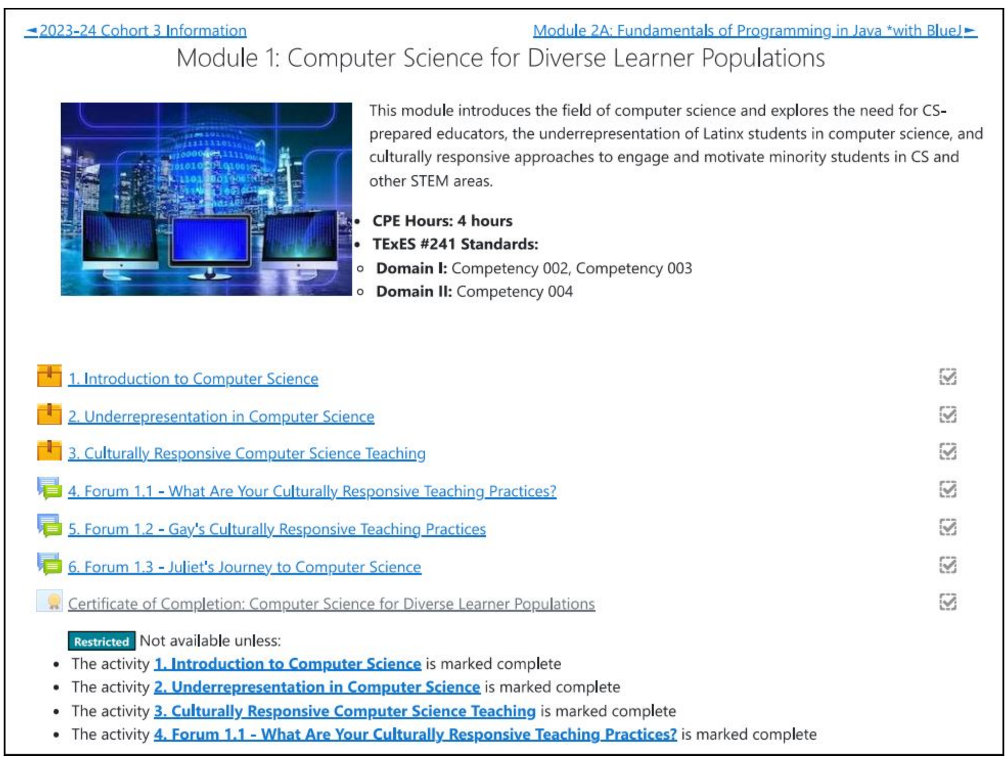

Figure 1: Sample from the CS4SA Computer Science for Diverse Learner Populations module.

The modules were designed to progress from foundational concepts and hands-on skills to more specialized topics, as defined by the TExES certification requirements [15], including:

- Programming Languages: Teachers explored Java and block-based languages such as Scratch to help them understand different approaches to teaching programming.
- Data Structures: Topics focused on key data structures, including arrays, lists, stacks, and queues, which are fundamental to computer science.
- Specialized Topics: The program also offered modules on robotics, digital forensics, discrete mathematics, and game/application development.

Figures throughout this section provide samples of specific modules and activities, which were reinforced by quizzes and interactive exercises. For instance, Figure 2 displays the CS4SA Data Structures module, which includes a coding exercise shown in Figure 3 where teachers created a card game in BlueJ to practice linked lists [16]. Other activities included building educational apps (e.g., a language translator or to-do list) and robotics projects using Arduino [17], micro:bits [18], and SPIKE Prime [19]. These activities were designed to reinforce the concepts introduced in the modules, providing hands-on practice alongside theoretical learning.

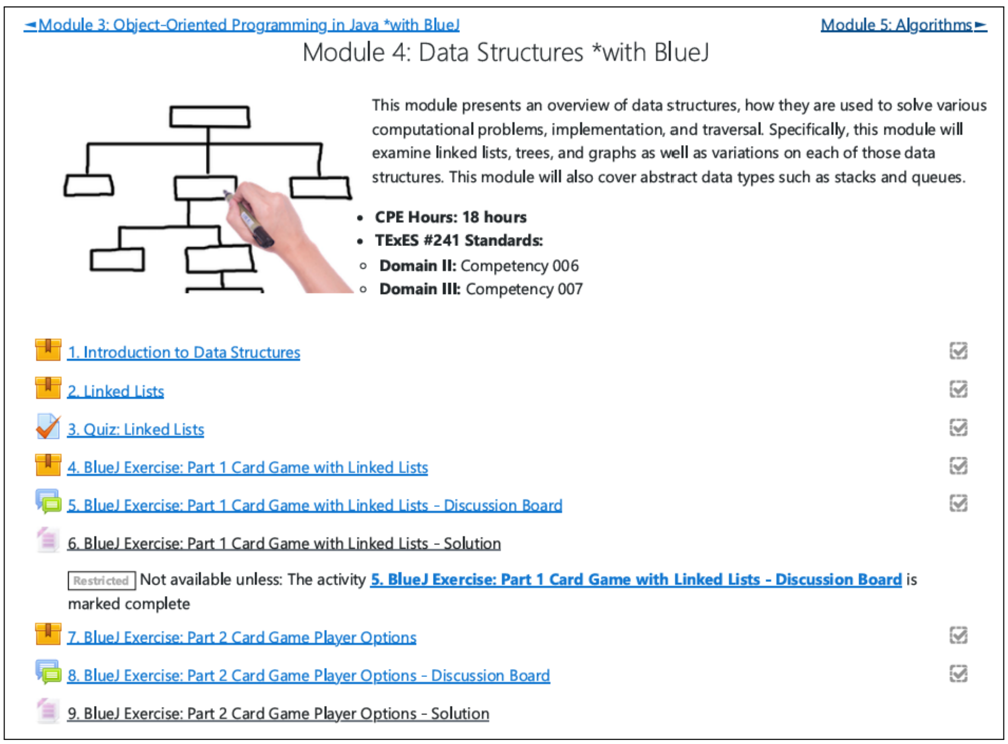

Figure 2: Sample from the CS4SA Data Structures with BlueJ module.

Building on programming and data structure foundations, participants explored algorithms such as recursion, searching, and sorting methods. The Algorithms module introduced bubble, insertion, selection, merge, and quick sorts, along with Big-O notation for analyzing efficiency. It also provided quick-reference Big-O tables. Figure 4 shows the home page of the module, outlining the resources and lessons available to participants.

Figure 5 illustrates a project from the Game Development with Unity and C# module, highlighting game and mobile application development. In this module, participants gained hands-on experience using industry-standard tools and processes. While the module did not focus

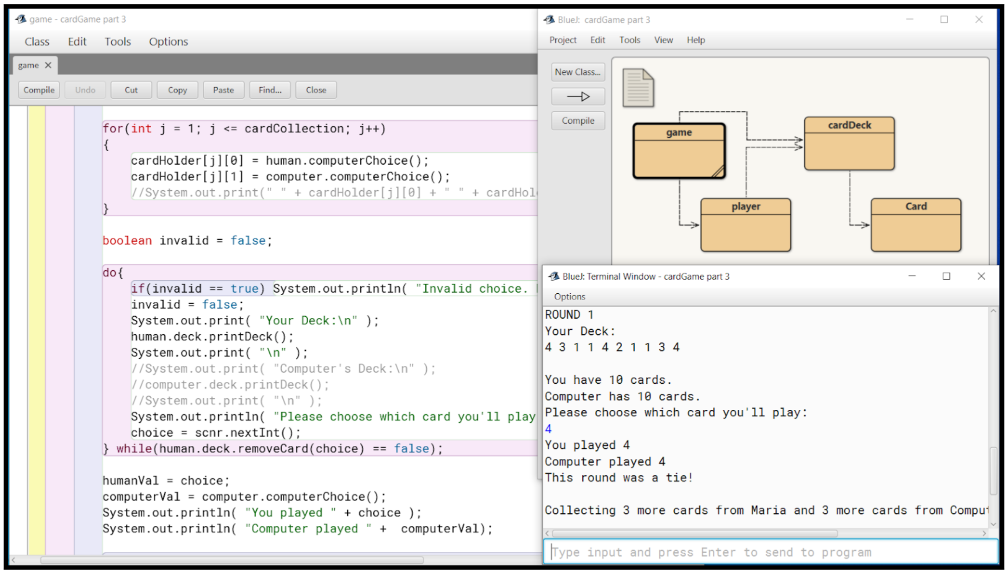

Figure 3: Sample Java programming project from the CS4SA Data Structures with BlueJ module, featuring card game exercises on linked lists created by our team.

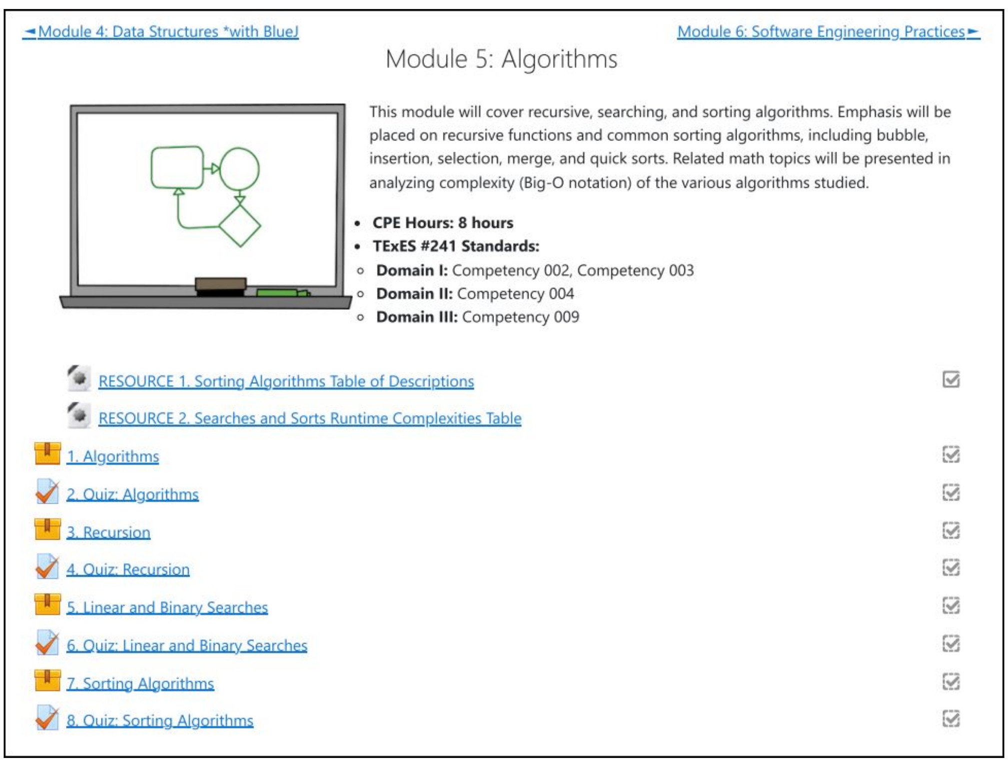

Figure 4: Sample from the CS4SA Algorithms module.

on teaching C# directly, it reinforced Java practices by emphasizing that C#'s syntax closely resembles Java [20]. This approach helped the teachers develop their programming skills while enabling them to integrate similar projects into their classrooms, creating engaging CS activities for their students.

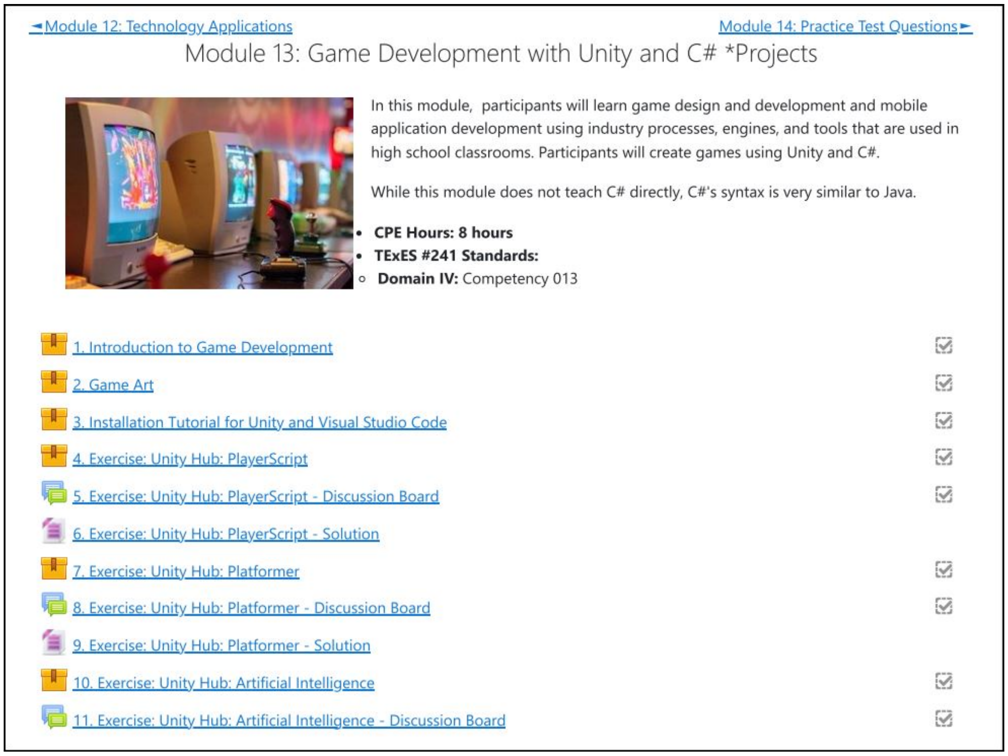

Figure 5: Sample from the CS4SA Game Development with Unity and C# module.

Hands-on, time-intensive programming projects were incorporated into several modules, marked with an asterisk (*). These projects were part of both the Summer Institute and the online modules. In addition to these projects, the program addressed a broad range of topics designed to equip participants with the skills necessary for teaching computer science. The following is an overview of the modules developed by CS4SA:

Summer Institute Topics:

1. Computer Science for Diverse Learner Populations
2. Fundamentals of Programming in Java (with BlueJ)*
3. Object-Oriented Programming (with BlueJ)*
4. Data Structures (with BlueJ)*
5. Educational Robotics (projects)*

Online Modules (after the Summer Institute):

6. Algorithms
7. Software Engineering Practices
8. Discrete Mathematics

9. Digital Forensics
10. Technology Applications
11. Making and Microcontrollers (projects)*
12. Mobile Application Development (projects)*
13. Game Development with Unity and C# (projects)*
14. CS Teacher Certification Test Preparation

Summer Institute with Youth Code Jam. CS4SA began with a three-week Summer Institute, where teachers were introduced to foundational CS topics, lessons, and culturally responsive pedagogy, alongside learning fundamental Java programming concepts using BlueJ, a beginner-friendly development environment [21]. Teachers engaged with the modules, completed programming exercises, discussed broadening participation in CS, and planned classroom activities. As shown in Figure 3, one of the early exercises in the Data Structures (with BlueJ) module involved developing a card game using linked lists, helping participants reinforce these foundational programming concepts.

The first week of the Summer Institute focused on introducing data types, structures, and functions in code development. The second week shifted to object-oriented programming and included a hands-on experience with a summer coding camp for secondary students from the teachers' school districts. The camp, run by Youth Code Jam (YCJ), a non-profit organization providing coding experiences to youth across Texas [22], was hosted at our university campus, the University of Texas at San Antonio. Teachers observed the YCJ instructors guiding students through block-based visual programming with MIT's Scratch [23]. These lessons reinforced CS terminology and control structures, helping teachers connect their learning with practical examples.

As part of the YCJ camp, CS4SA provided participants with micro:bits, small devices equipped with embedded sensors that can be programmed using block-based coding similar to Scratch [18]. Robotics, a specialized topic for CS certification [15], was integrated into the camp activities through these devices. Participants learned to program the micro:bits to collect and display sensor data and create interactive games. After completing the activities, students and teachers were allowed to keep the micro:bits, enabling continued exploration and application of these concepts.

Inspired by the students' engagement, YCJ instructors included a day working with their set of Finch robots. These robots, developed at Carnegie Mellon University, are designed to support educators in hands-on CS and STEM projects, particularly for underserved communities [24]. YCJ also informed our teachers about the Finch Loan Program, a free initiative that allows schools to borrow a set of robots for approximately two months [25]. Finch robots, similar to the micro:bits, use block-based programming but are larger devices capable of driving around when programmed. The activities with both micro:bits and Finch robots reinforced real-world CS applications and coding concepts. One teacher utilized the Finch Loan Program to incorporate the robots into after-school activities.

The YCJ camp, attended by secondary students recruited from the teachers' school districts through our program, provided an immersive experience for teachers and students. Teachers participated in programming activities alongside the students, learning to develop their own CS lessons by observing the experienced YCJ instructors. The camp focused on block-based programming, with Scratch, micro:bits, and Finch robots all using this approach. These hands-on activities reinforced key CS concepts and allowed teachers to connect their learning with real-world applications, preparing them to implement similar classroom activities.

Monthly Workshops. After the Summer Institute, CS4SA continued with six monthly workshops. CS4SA workshops featured modules on topics including Digital Forensics, Discrete Mathematics, and Cybersecurity Fundamentals to provide foundational CS knowledge aligned with high school standards such as CS Principles [26] and AP Computer Science A [27]. At the same time, other modules focused on real-world applications and hands-on projects to inspire engagement and broaden the appeal of computer science for diverse student interests.

Cohort 1 workshops were held online for 1.5 hours, while Cohorts 2 and 3 participated in 6-hour in-person meetings. These workshops allowed teachers to troubleshoot issues, discuss their experiences implementing CS lessons, and engage in hands-on activities. Teachers worked through more advanced topics from the online modules, revisiting challenging concepts, reviewing practice test questions for the TExES 241 exam, and participating in group discussions.

Collaborative Activities and Peer Mentoring. As the program evolved, CS4SA increased opportunities for collaboration and peer mentoring. Teachers used discussion boards to share solutions, ask questions, and exchange resources. Weekly emails, introduced at the suggestion of participants, featured a coding question-of-the-week linked to discussion board posts for further engagement. Teachers were encouraged to show their work when solving problems and to ask questions as needed. These emails also highlighted upcoming STEM activities, such as robotics competitions, helping to maintain engagement between sessions.

The monthly workshops provided opportunities for discussions, troubleshooting, and collaborative activities, including hands-on, unplugged CS projects. By the third cohort, teachers began working through the modules together during these sessions, transitioning from the initial asynchronous format. The workshops also provided dedicated time to review challenging practice test questions from the TExES 241 resources [28], offering an opportunity to analyze answers and strategies as a group. Over time, online tutoring sessions became more structured, with group reviews and additional resources to support test preparation.

Balancing Standards with Inspiration. Modules such as Algorithms, Software Engineering Practices, and Discrete Mathematics provided foundational CS knowledge aligned with high school standards [15], and modules such as Educational Robotics, Digital Forensics, and Game Development with Unity and C# emphasized hands-on, real-world applications. Robotics, for example, allowed teachers to engage in block-based programming through Finch robots, demonstrating the practical impact of CS. Other modules, such as Mobile Application Development and Making and Microcontrollers, inspired creativity and problem-solving through tangible projects.

Field trips to the university provided an experiential component, connecting academic content to potential career paths and higher education opportunities. These sessions highlighted the broader relevance of CS, encouraging teachers and students to see its real-world applications and future possibilities.

More Hands-On Robotics. Building on the hands-on approach of Finch and micro:bits, CS4SA expanded the robotics experience with additional kits, including SPIKE Prime [19] and Arduino kits [17]. These robotics lessons were designed to build foundational skills gradually and progress to advanced challenges such as line following, obstacle navigation, and object throwing. Workshops introduced SPIKE Prime through hands-on tutorials using simple materials, such as craft supplies, while independent modules with embedded videos allowed teachers to continue learning at their own pace. Figure 6 shows a sample SPIKE Prime tutorial with video overlays for step-by-step guidance. Bonus training videos and additional exercises supported further learning and prepared teachers for classroom implementation. The hands-on, real-world applications found in PBL are essential for developing independent thinking, critical analysis, and collaborative learning with peers. Dema & Choden [29] emphasize that PBL is more valuable than traditional instruction because it allows students in the classroom to experience the authenticity embedded in real-world projects.

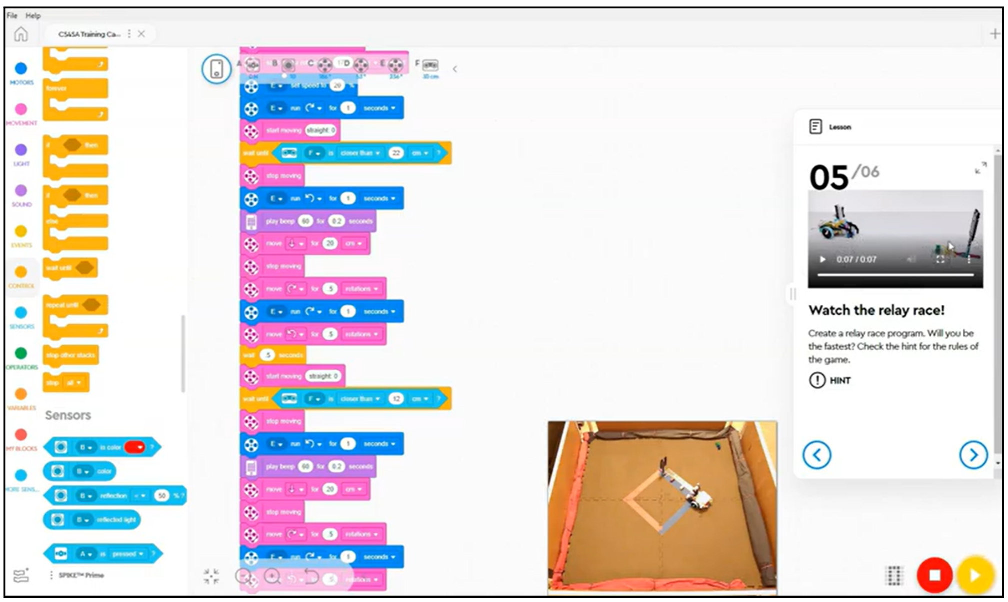

Figure 6: Sample SPIKE Prime tutorial from LEGO Education [19], enhanced with additional content from CS4SA, featuring step-by-step video guidance.

Advanced Coding Projects. CS4SA leveraged free online educational resources, such as MIT App Inventor tutorials, enhancing them with additional exercises, classroom tips, and real-world applications. These resources supported teachers in preparing students for future opportunities in higher education, careers, or general CS learning. For instance, the Mobile Application Development module utilized App Inventor, a beginner-friendly block-based programming tool [30], to guide teachers through projects such as a language “Translation App” and “Artificial Intelligence with App Inventor” [31]. These expanded resources also encouraged participation in initiatives, including the Congressional App Challenge [32].

Technical Challenges with App Inventor. Teachers faced technical challenges with App Inventor, including issues displaying projects on emulators or mobile devices. Although troubleshooting resources were available online, the time required to resolve issues sometimes discouraged users.

Nevertheless, App Inventor's robust tools provided valuable opportunities for creating advanced CS projects.

Technical Challenges with BlueJ. BlueJ, introduced earlier for Java programming, presented installation hurdles due to IT restrictions on school devices, limiting classroom use. Teachers were encouraged to explore BlueJ on personal devices or collaborate with IT departments to facilitate installation, as it remains a valuable tool for teaching Java concepts.

Making and Microcontrollers. For CS4SA's module Making and Microcontrollers, free online Arduino resources were adapted and enhanced with custom tutorial videos, classroom tips, and real-world applications. Teachers completed projects such as converting temperatures between Celsius and Fahrenheit, displaying sensor readings on LED devices, and building a motorized pinwheel. These projects introduced text-based coding similar to Java, as well as hands-on experiences that teachers could share with students. Arduinos were chosen due to COVID-19-related supply chain issues with micro:bits, which were originally planned for the module.

Technical Challenges with Arduino. Using Arduinos posed several technical challenges. Assembling the numerous small wires and components was difficult, and the instructions were hard to follow due to mismatched wire colors, requiring teachers to consult a separate chart. Additionally, the Arduinos were prone to falling apart easily, and several units had malfunctioning LED displays. For teachers considering working with Arduinos beyond the introductory projects, we recommend using an online simulator such as Tinkercad's Arduino simulator before building the physical device [33].

Unity and C# Software in Game Development. The program also leveraged free online Unity and C# resources, which were expanded with custom videos and tips to guide teachers through industry-level game development projects [34]. These projects built upon the block-based programming foundation of earlier modules and transitioned participants to text-based coding, building confidence in advanced programming concepts. Unity's professional-grade tools allowed teachers to explore real-world applications of game development, equipping them with skills to inspire students interested in careers in the industry.

Insights: Pilot and Cohorts

The CS4SA pilot program and three cohorts offer insights into the evolution and impact of the PD initiative. CS4SA partnered with a large urban school district comprising 98 campuses, serving a predominantly low socioeconomic demographic. Statistical data indicates that 79% of students identify as Hispanic or Latinx, 7% as Black or African American, and 36.4% of families have incomes below the poverty level, compared to the national average of 14.2% [35]. Teachers participating in the program reflected similar demographics, with many coming from schools serving predominantly Hispanic and Latinx populations. Teacher participants taught various subjects, including math, science, business, engineering, and information technology (IT), along with alternative school teachers and a librarian.

The pilot and Cohort 1 operated entirely online due to the COVID-19 pandemic, while Cohorts 2 and 3 included a mix of in-person activities and online module work to complete outside of scheduled sessions. Feedback from each cohort informed program adjustments, particularly in response to the challenges posed by the pandemic.

Summary of Cohorts:

- Pilot Phase (Aug 2020 - July 2021): 15 participants started online; 3 moved on to Cohort 1, and 1 joined Cohort 2.
- Cohort 1 (July 2021 - June 2022): 12 participants started online; 8 completed the program, and 1 teacher passed the CS teacher certification exam. Total Cohort: 8 participants.
- Cohort 2 (July 2022 - July 2023): 4 participants started (online and in-person); 6 mentors from Cohort 1 joined, 2 completed the program, and 1 teacher passed the CS teacher certification exam by the end of Cohort 3. Total Cohort: 8 (participants + mentors).
- Cohort 3 (June 2023 - September 2024): 3 participants started (online and in-person); 3 mentors from previous cohorts joined, 3 completed the program. Total Cohort: 6 (participants + mentors).

Table 1 provides a summarized overview of key data points from the pilot phase and the three cohorts, offering a clearer comparison of the cohort details.

Table 1: Summary of Cohorts

Cohort	Dates	Mode	Started	Mentors	Completed	Certified	Total Cohort
Pilot	2020-21	Online	15	-	-	-	-
Cohort 1 (C1)	2021-22	Online	12	-	8	1 (2022)	8
Cohort 2 (C2)	2022-23	Hybrid	4	6 (C1)	2	1 (2024)	8
Cohort 3 (C3)	2023-24	Hybrid	3	3 (C1 & C2)	3	-	6

The following insights present the experiences of the pilot and each cohort, exploring their challenges, adaptations, and outcomes. The project received Institutional Review Board (IRB) approval for the research activities associated with this program, and informed consent was obtained from the teacher participants for data collection.

Data Collection. Data were collected through semi-structured interviews with participants at the program's start and end, along with their postings on shared documents, discussion boards, and other submitted artifacts (e.g., code exercises). These qualitative data gathered throughout the program, including online workshops for cohort 1 and in-person workshops for cohorts 2 and 3, offer valuable insights into the growth of teachers in pedagogical practices, content knowledge, and overall experiences during the professional development program.

Future research would benefit from additional documentation, including surveys administered to teachers at the end of their participation. While grade levels and subjects taught by participants were collected verbally, formal documentation of this data through surveys would benefit future research. Having this information recorded in a structured manner would provide valuable context for understanding participants' specific needs and allow for more targeted recommendations in subsequent programs.

Pilot Program

Initially intended as an in-person program, the CS4SA pilot phase (August 2020 - July 2021) was adapted to an online format due to the pandemic. Our team began developing the program modules in August 2019, and the pilot phase, known as CS4SA-NOW, delivered half of the planned content. Due to the disruptions caused by the pandemic and the early stage of the program, data collection on this cohort was limited, and formal interviews with participants had not yet begun. Additionally, the pilot cohort did not participate in the Youth Code Jam Summer Camp, which began with Cohort 1. Of the 15 teachers who began, three completed the program and advanced to Cohort 1. A fourth teacher from the pilot program joined Cohort 2 but did not complete the program until the end of Cohort 3, at which point they successfully passed the CS certification exam.

Cohort 1

Overview. Cohort 1 (September 2021 - May 2022) was the largest cohort, likely due to the flexibility of the fully virtual format, which removed the logistical barriers of in-person attendance. While the remote setup presented challenges, such as difficulty for some instructors with the online format, it also offered much-needed flexibility during the pandemic. For many teachers, participating at their own pace allowed them to balance work commitments more effectively, contributing to a positive experience despite disruptions. The cohort began with 12 participants, 8 of whom completed the program, with one passing the CS teacher certification exam. Most teachers had non-CS backgrounds, primarily teaching math, with a few in STEM, engineering, and business. One teacher, already CS-certified, participated in CS4SA for networking opportunities and later served as a mentor for Cohort 2. Several teachers remained with the program across all three cohorts. One math instructor struggled with the online format, completing only half of the program before returning to Cohort 3 to finish after the transition to in-person professional development.

Youth Code Jam Camp. Cohort 1's YCJ camp was conducted entirely online, with one exception. Approximately 20 high school students, enrolled in a summer class supervised by one of our teacher participants (a math teacher), attended the morning YCJ sessions in person with their teacher. However, the camp was virtual, and the teacher participants engaged remotely. The students worked on Scratch programming, while the teacher participants followed along virtually to learn how to teach CS activities. In the afternoon, with low student participation, the remaining teachers worked with YCJ instructors on CS4SA's Java and programming modules.

Workshops. During the COVID-19 pandemic, monthly workshops for Cohort 1 were initially held online and limited to 1.5 hours due to the extended hours teachers in our cohort had already spent on Zoom for their own hybrid or virtual classrooms. In a hybrid classroom, some students attended in person while others participated online, requiring teachers to manage both formats simultaneously. To avoid screen fatigue, the session duration was kept shorter. However, for Cohorts 2 and 3, as in-person workshops and classes resumed, the duration was extended to 6 hours to allow for more hands-on activities and deeper discussions. Optional tutoring sessions were offered for deeper CS knowledge and exam preparation, and although attendance improved, the sessions became more effective when scheduled closer to the certification test date.

Participant Experiences from Interviews and Observations. The following insights are based on interviews and observations conducted with participants throughout the program, starting with Cohort 1. The online format of Cohort 1 provided flexibility, which was particularly beneficial during the pandemic, allowing teachers to meet remotely and share resources and experiences related to CS. Teachers worked on projects in robotics, game and app development, and Java, while learning introductory CS concepts to explain CS to their students.

A component of CS4SA was integrating culturally responsive teaching to help teachers connect with their students and contextualize CS meaningfully. From the summer institute to the first module, discussions encouraged them to reflect on their students' backgrounds, interests, and experiences. One teacher shared how these discussions shaped their perspective and approach to the classroom:

> Culturally responsive teaching has, I really think, made an impact on me. I'm getting emotional. (Laughs.) On me seeing my students in a different way, like it seemed so simple, and I don't know if, it's probably because COVID was a precursor to it, right? But COVID put us all on the same playing field. We all went through this together, then we had snow-VID [36], that we all went there together, and so like just hearing that I can spend the first five minutes of my class saying, 'Okay, guys, what's going on? What are you up to? You can talk or you don't have to talk or...' Because I felt like I would 'always relate to my students somehow.' I put that in air quotes, cause I like football and I like basketball and I can relate to them on that, you know. But this year the amount of conversations that were more personal, is just really special. I'm a crier, excuse me. (Laughs.) But really, and it's, what I like about that part of it is that we weren't always saying, 'We're now talking about culturally responsive teaching. Now, this is the part about culturally responsive teaching...' You know? But that part was just really special to me, and it helped me grow, I think, as a teacher. And then of course, I mean I like nerding out. That's why I'm a Robotics teacher. I like learning. So everything, you know, that we did was great. It was hard. (Laughs.)

An important aspect of Cohort 1 was the use of engaging activities that helped teachers encourage student involvement and support their teaching practices. One IT teacher shared his experience with the tools provided by CS4SA and how they made a significant difference in student engagement:

> I think what I did very different than other years that I've taught, I mean I've only taught [CS-related activities such as programming] for three years, and so what helped a lot was the toys that we got. I call them toys, but engaging activities. That helped a lot, in carrying the class a little bit more, with a little more engagement than other years. You know, the robotics with the Lego and the Arduino sets, kind of does activate some engagement from the students, so I think that overall helped me a lot in getting through the class, and not only in engaging the students, but also in giving me a little break on the regular routine of the programming.

Beyond the impact on individual teaching practices, the program also reinforced the importance of advocating for expanded CS opportunities. Another goal of CS4SA was to cultivate CS advocates within the district, which led to the inclusion of a CS-certified teacher in the program. Initially unable to enroll in the pilot due to his certification, he was later invited to join as the program sought to recruit more teachers. Motivated to expand CS access, he pursued networking opportunities and explored ways to create CS learning experiences in his district. Although certified, he encountered limited demand for CS teachers and began his role teaching math. To introduce CS in his classroom, he incorporated CS concepts into lessons, such as having students create websites. His involvement in CS4SA further strengthened his advocacy, and he later served as a mentor for Cohort 2, sharing resources and supporting other teachers in integrating CS into their instruction.

Outcomes. While there were challenges, CS4SA yielded several positive outcomes. Teachers valued the opportunity to learn about CS and found it useful for their classrooms. One teacher noted how improving listening skills helped to strengthen connections with students. The program also facilitated a community where teachers could network and share ideas.

Another math teacher with a science background made CS more relatable by inviting his brother, a software developer for a local grocery chain's mobile app, to conduct an online session for his students. He integrated real-world CS examples into his math curriculum. In addition, he attended board meetings to advocate for CS to be included for the district's STEM stipend, which was available for math but not for CS. As he explained:

> My goal was to try and help get computer science, CS teachers, the STEM stipend because I thought it was ludicrous that CS teachers don't receive the STEM stipend. I mean, what is more STEM than CS? So, I had written a plan of going to the Board, and I would say I didn't get to fulfill that, but I did gain a little bit more knowledge. I did attend a board meeting and kind of learned a little bit more about the background process of signing up for Citizens Comment, which is essentially the outlet to express that want or concern. I guess between having a switch in superintendent this year, we had an interim superintendent the entire year except for this last month, and we finally got our permanent one. That made it a little bit more difficult, but honestly, I was kind of just dismayed by our Board, and seeing how little they're willing to truly listen to the community and make change.

This discovery was both surprising and disheartening. Despite his advocacy, budget constraints made it difficult to enact change, highlighting systemic barriers to CS education. The lack of financial support for CS teachers became a discouraging reality, dampening enthusiasm among some participants who had considered certification or deepening their CS expertise. What began as an effort to expand CS opportunities ultimately underscored the challenges of institutional change, leading some teachers to question the feasibility of long-term CS integration.

To support teachers in their professional development, the program offered reimbursement for up to two attempts at the CS teacher certification exam, upon submission of receipts. One teacher, who taught business and math, successfully passed the exam after studying the program's materials and engaging with coding exercises. While passing the exam was a measurable success, many teachers explored and implemented engaging CS activities in their classrooms. Teachers integrated CS concepts in various ways, such as through after-school clubs, free-time activities, or elective courses. Some used program resources to support projects like Unity game development, robotics, and Scratch programming. These hands-on projects helped teachers engage students and create real-world connections to CS. Integration was flexible, with teachers adapting it to different grade levels and class structures. In particular, the teacher with a business and math background introduced CS through both formal math lessons and informal after-school clubs, aiming to make CS more accessible and relevant. This approach not only engaged students but also helped teachers connect with CS content and explore how to incorporate it across subjects.

Cohort 2

Overview. Cohort 2 (July 2022 - July 2023) faced challenges with teacher participation and student enrollment. Despite district recruitment efforts, the low turnout for our program was unexpected. Possible contributing factors include the return to in-person classes, increased emphasis on state testing, and the perceived substantial commitment required for professional development hours. The cohort primarily consisted of teachers from IT and math backgrounds, with six mentors from Cohort 1, mostly math teachers but also an alternative school teacher, helping sustain engagement throughout the cohort. Additionally, a university presentation was incorporated into the summer camp, where participants learned about degree programs offered, including those in CS, giving both teachers and students insight into potential higher education opportunities.

Workshops. The cohort employed a hybrid model, combining online content with in-person six-hour workshops. These workshops provided hands-on programming exercises, opportunities for teachers to share experiences, and addressed challenges in CS education, while supplementing the online materials with a collaborative learning environment. Weekly emails, a suggestion from a returning teacher mentor, provided coding challenges and updates on STEM activities. These emails helped maintain engagement and encouraged continued teacher interaction through discussion boards.

Participant Experiences from Interviews and Observations. CS4SA offered opportunities for teachers to network and share CS activities and resources. Due to the time-consuming nature of the modules, the team shifted to more direct engagement with the CS lessons based on feedback midway through the program, in contrast to the primarily asynchronous approach in Cohort 1. The transition back to in-person learning also presented challenges, as teachers and students navigated new expectations and the lingering effects of pandemic-related disruptions. One participant reflected on the exhaustion and shifting priorities during this period:

> Everybody is definitely tired, and I think last year was like, kind of like we were still in like, ‘Oh, my gosh!’ Maybe they were in [CS4SA] because like computers, during the COVID or the pandemic, we had to rely on them more, and this year was kind of like, ‘Yes, we’re tired,’ but it wasn’t such a necessity, maybe, to understand different things, and if you don’t really know what all entails with the computer science or the course. Like, you’re like, ‘Computers, I need to learn more about computers,’ so I don’t know. That could be a possibility, too… A lot of the students were virtual, right? You know, a majority of the students were virtual so in the spring of that year, I

> feel like okay, it was like a learning curve, like, 'Oh, my gosh, like rush and let's do this.' The next year we're like, okay, we started off, it was very like, blended, some students stayed home, some came. It was kind of like, 'It's okay.' But then this year was the expectation back to normal… so everything that was expected before is expected now. However, the kids, most, a lot of them hadn't been in school for a year and a half, or a year, and so it's a lot of like kids that were behind or even further behind, you're teaching them how to, it sounds cliche, but like how to learn school again, but it's really all of that trauma and all of those experiences that they're coming back with and, yeah, everyone was tired. I think there was low energy because there we had, the state didn't say, 'Oh, okay, you know, I know the kids are coming back with these deficits or these issues, you know, don't worry about STAAR [State of Texas Assessments of Academic Readiness exam] [37], don't worry about this.' For a lot of schools like us, we were an IR [Improvement Required] [38], so we had to push and dig and dig and dig to get out of IR. You know? So everyone's tired, I think.

One teacher emphasized the need for programs like CS4SA, noting the shortage of CS-certified educators in their district. He expressed concern that without a pipeline of trained teachers, schools would struggle to sustain CS course offerings, stating, "You can't expect your district to offer computer science courses if none of your people are computer science-certified." He also highlighted the challenge of teacher retention, explaining, "If you have only three people certified in your district and one leaves, the program goes with them." To address this, he stressed the importance of administrative support and incentives, saying, "The school district should offer a stipend to keep people in that training too... and provide whatever they need so that this one certification, you have a steady stream of people." Ensuring a sustainable path for CS educators would help schools expand their course offerings beyond a single introductory class and build long-term CS programs.

Outcomes. By the end of Cohort 2, two teachers had completed the program. The math teacher incorporated robotics activities into his curriculum. The IT teacher used Python [39], a language similar to Java, to engage students in creating simple game projects. During Cohort 2, he completed all in-person requirements but only half of the online components and did not take the certification exam. Because he showed strong potential to pass the challenging certification exam, he was encouraged to join Cohort 3 study sessions, where he actively participated in online sessions and study groups. As a result, he completed the program and became the only participant across Cohorts 2 and 3 to pass the CS teacher certification exam. Although the exam was difficult, the IT teacher noted that certification would open opportunities for him to teach more CS in the future. After passing the exam, he successfully added AP Computer Science Principles to his teaching schedule, along with other IT courses.

Cohort 3

Overview. Cohort 3 (July 2023 - September 2024) included three teachers supported by three mentors with engineering, IT, and math backgrounds. Recruitment for the cohort was challenging, potentially due to factors such as the district's declining enrollment [40], [41], reduced teacher capacity during COVID-19, and layoffs as pandemic-related funding expired [42]. These challenges may have contributed to the low participation rate. Recruitment for the YCJ camp was initially difficult, but by the program's end, approximately 20 students had joined.

CS4SA provided micro:bits to spark students' interest in coding, which the teachers and students could keep. During the YCJ camp, teachers and students participated in lessons involving both micro:bits and Finch robots, where the YCJ instructors informed them of the Finch loan program. Following the Summer Institute, teachers implemented similar activities with their students early in the school year before testing demands took precedence. As with Cohort 2, Cohort 3 participated in a university presentation highlighting career opportunities in computer science and STEM fields, which teachers found valuable.

Workshops. In Cohort 3, our team adopted a more structured approach, incorporating the modules more extensively into in-person meetings than previous cohorts. This adjustment, informed by feedback from earlier cohorts, aimed to enhance engagement by emphasizing interactive and synchronous experiences. One teacher began the summer portion online to accommodate recruitment flexibility, but once the school year started, all teachers participated in monthly in-person workshops. While module work remained primarily online, the Summer Institute and workshops offered greater opportunities for collaborative, in-person engagement, which teachers and mentors found valuable.

Participant Experiences from Interviews and Observations. In Cohort 3, a high school mathematics teacher, inspired by CS4SA's robotics lessons, integrated computer science into his curriculum using robots and graphing calculators. He introduced students to coding by programming the Texas Instruments Innovator Rover [43] to move the calculator as a rover robot, utilizing Python, a language similar to Java. He also designed coding exercises to solve geometry and trigonometry problems. Despite technical challenges and time constraints, the teacher valued Python's ease of use and remained committed to incorporating coding into his lessons.

In addition to these efforts, the teacher saw the positive impact of culturally responsive computer science pedagogy on student engagement. Reflecting on this, he shared:

> When we were using Scratch, I told my students to do a project on anything they wanted to do, things they like, things that are part of their culture, and they created this fictional character that resonated with their culture. All the students were laughing, having fun, and they were having a good time so it made me see how we can learn and enjoy what we're doing at the same time.

A high school librarian introduced students to Finch robots through the Finch loan program [25] she discovered at the YCJ summer camp. Hands-on activities in our CS4SA program gave her the confidence to lead robotics activities such as robot games, line following with sensors, and obstacle detection. Although technical limitations such as locked-down Chromebooks posed challenges, the librarian creatively adapted activities and engaged students through competitive robot challenges. She observed that many students were unaware of the broader opportunities in CS, emphasizing the need for greater awareness.

> I was shocked that more of our students weren't aware of the things that you could do with computer science, even though we have, we call it CATE, which is Computer And Technology Education, so we have classes already in the high school, but they focus mostly on getting certified for, like, Microsoft… I'm not sure that they know

> that [CS] is there and what they can do with it once they leave because we have the firefighter program, we have a CNA [Certified Nursing Assistant] program, we have a pharmacy program, we have a dental assistant program. All those programs are pretty concrete to them, so then you have the CATE program competing with these other ones, and I think it's a marketing issue; just getting the information out there to the students to let them know.

Despite these challenges, she remained enthusiastic about integrating coding into her new middle school library and makerspace.

A teacher from an alternative school worked with students for shorter durations and integrated coding into her math lessons using robots. Although state testing and attendance challenges limited her time, she incorporated the robots into activities related to geometry and angles. Inspired by these activities, she sought a grant early in the school year to acquire more robots for her classroom. She appreciated the value of CS and planned to continue integrating it into an after-school program at her new middle school, where she hoped to develop student-created games for State of Texas Assessments of Academic Readiness (STAAR) exam preparation.

Outcomes. Teachers in Cohort 3 faced several challenges, including technical difficulties, scheduling conflicts, and the pressures of state testing. Some software tools, such as BlueJ and Unity, were difficult to install on school computers, limiting their usability. In contrast, online tools that did not require installation proved more successful. Even with these obstacles, the cohort achieved positive outcomes, and teachers expressed a desire to continue incorporating CS into their classrooms.

The math teacher, who had started with the pilot program and participated in Cohort 1 but did not complete it, enrolled again in Cohort 3, believing that the in-person version of the program would lead to greater success. He plans to continue using Python in his lessons. The alternative school teacher, who completed the program except for the certification exam, is transitioning to a middle school math role and aims to develop after-school CS clubs and integrate coding into other subjects such as math and science. The librarian plans to expand her CS activities such as robotics in her new middle school makerspace, and highlighted the broader impact of her own learning experience, stating, "I got more knowledge about computer science in general, and being able to share it with my students, whether they're in high school or middle school, was very positive… It was good to learn something new and good to be able to share it with the students. There were no negatives at all."

Additionally, the IT teacher from Cohort 2, who had initially opted not to take the certification exam, was invited to join Cohort 3's study sessions due to his firm grasp of the content. He actively participated in online discussions, shared knowledge with other participants, and practiced test material. The recorded study sessions were made available for review, benefiting the entire cohort. After gaining confidence through these sessions, he passed the certification exam, and as a result, now teaches AP Computer Science Principles along with other IT courses.

Discussion

These insights demonstrate how teacher participants took what they learned from the professional development program and created new CS and other STEM experiences in their classrooms. Professional development aims to lead to changes in teacher practices [4], [5]. In this program, we supported teacher participants in implementing activities and lessons in a new content area, which they then attempted to incorporate into their existing curricula. Across these three cohorts, teachers expressed excitement about integrating CS into their teaching despite the challenges posed by limitations in existing curricula, resources, and time.

All teachers successfully implemented some computing activities, such as programming in their math courses and incorporating programming and robotics into students' free time. They also reported that the creativity and competitive aspects of the coding activities were particularly engaging for their students. In addition, the teacher participants expanded their activities to include new technologies (e.g., Finch robots, TI Rover) that were not addressed by CS4SA. As a result, the district had to approve these new technologies, and external technical support was required to assist with their implementation.

A recurring theme among the teachers was a need for improved district support to provide sufficient computing resources and access to relevant software programs. In non-traditional settings, such as libraries or alternative classes, teachers found that planning activities around free time, avoiding peak testing periods, and preparing CS and coding exercises in advance helped mitigate scheduling and curriculum constraints. Though they were able to acquire resources and create CS opportunities for their students, there were some challenges in terms of district and institutional barriers. With a priority on testing, integrating CS into one of those test areas made it somewhat easier to create more CS opportunities, as noted by the math teacher. For the librarian and alternative school teachers, the primary challenge was scheduling, as their instructional time falls outside regular class periods due to their roles. Since CS is not a tested subject, these activities are often conducted after school through clubs in public schools [44]. For more intentional integration, additional support from districts, as well as the principals and curriculum leaders, is necessary.

Another significant challenge highlighted by the cohorts was technical support. CS4SA collaborated with the school district to ensure teachers could access the required technologies (e.g., Integrated Development Environments). Despite these efforts, teachers encountered difficulties obtaining the correct access to the technologies and faced general challenges using the software. Peer mentors from previous cohorts played an important role in offering advice on navigating these institutional obstacles. These challenges underscore the need for ongoing support and are addressed in the following section.

Recommendations and Conclusion

This paper presented insights from three cohorts of in-service secondary teachers participating in a professional development program designed to prepare educators to teach computer science in various classroom settings. CS4SA blended instructional content with project-based activities, enabling teachers to integrate computer science into their teaching practices. Following the program, teachers implemented CS experiences in their classrooms, helping to engage a broader range of students in computer science learning. The findings highlight the potential of professional development programs to inspire teachers to expand CS offerings and address challenges in integrating new content into existing curricula and extracurricular activities.

Reflecting on participants' experiences during the past pilot and cohorts, several instructional recommendations are offered to other educators and researchers attempting a similar CS PD program for in-service teachers new to CS. It is essential to acknowledge that CS is a challenging subject to learn. While our project provided ample practice opportunities throughout each cohort, teacher participants needed additional practice with a range of problems scaffolded to their development. This approach aligns with the worked examples principle from cognitive load theory. Nearly a decade ago, Skudder & Luxton-Reilly [45] urged greater adoption of worked examples in CS and called the practice a "signature pedagogy for Computer Science." In addition to worked examples, participants would benefit from consistent, timely, and tailored feedback on their implementations of CS concepts.

While grade levels and subjects taught by participants were collected verbally, formal documentation of this data through surveys would be beneficial for future research. Having this information recorded in a structured manner would provide valuable context for understanding participants' specific needs and allow for more targeted recommendations in subsequent programs.

Another key recommendation is a greater emphasis on CS state exam preparation. The final quizzes in CS4SA were modeled after the CS state exam preparation guide [15], [28]. However, the state exam posed more challenging and critical application questions, encompassed topics beyond the preparation guide, and was difficult for participants to complete within the allotted time. In our final cohort, we addressed this gap by realigning the final preparation quizzes and including more exam practice with new module-specific quizzes. It is also worth noting that teachers should be encouraged to not only practice the quizzes frequently throughout the program but also time their exam preparation closer to the exam date, with adequate repetition and the ability to explain challenging concepts succinctly. Participants who practiced the preparation quizzes more frequently and closer to their exam were better prepared than those who practiced less. Additionally, incorporating test prep materials similar to those used in AP CS courses, such as CS A and Principles, would improve preparedness, especially for code tracing exercises.

The participants' experiences highlight further areas for improvement and potential growth in future PD programs. One crucial area is increasing CS advocacy and demonstrating the relevance of CS in diverse classroom settings. Teachers reported challenges in teaching CS due to students' general lack of awareness of the subject, which may contribute to low demand for CS courses. To enhance teacher engagement and motivation, we recommend offering more hands-on, interactive experiences, along with additional support for teachers outside formal sessions. Teachers strongly valued the opportunity to start modules together in person, as it allowed for immediate troubleshooting of technical issues and collaborative problem-solving. They then appreciated the flexibility to finish modules at their convenience, since all content was available online. Increasing in-person learning time would help create a stronger sense of community and provide more personalized support for integrating CS into classrooms. Additionally, providing teachers with practical, culturally responsive CS activities could help students see the relevance of CS, increasing engagement and ownership of their learning.

Online learning modules played a significant role in facilitating progress and flexibility, allowing teachers to engage with content at their own pace. However, the preparation and maintenance of the program required substantial effort to ensure the quality and relevance of the materials. To improve future offerings, CS4SA could consider shortening certain components, such as making the Arduino module optional and reducing the game development module, while increasing in-person hands-on projects to address technical challenges that are difficult to resolve asynchronously. Switching to an online IDE that minimizes installation issues could also improve the experience of teachers, such as using an alternative to BlueJ, which presented setup challenges.

By creating opportunities for teachers to explore CS in meaningful ways, the program demonstrated the potential to inspire teachers and their students. Teachers valued hands-on learning experiences, opportunities for collaboration, and the flexibility of hybrid instruction. Many participants expanded their CS teaching, introduced new projects into their classrooms, and increased student interest in computing. With more targeted support, advocacy, and an emphasis on the relevance of CS in diverse educational contexts, future initiatives can build on this work and continue to broaden access to computer science.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1923269. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Stuart Reges and Marty Stepp. *Building Java Programs: A Back to Basics Approach*. Pearson, Inc, Hoboken, NJ, 2020.

[2] Geneva Gay. *Culturally Responsive Teaching: Theory, Research, and Practice*. Teachers College Press, New York, NY, 2018.

[3] Timothy T Yuen, Emily P Bonner, Maria G Arreguin-Anderson, Anne E Campbell, and MS Trevisan. *(Under) Represented Latin@s in STEM: Increasing Participation Throughout Education and the Workplace*. Peter Lang Publishing, Inc, 2018.

[4] L Darling-Hammond. Effective teacher professional development. *Learning Policy Institute*, 2017.

[5] Gwen Nugent, Keting Chen, Leen-Kiat Soh, Dongho Choi, Guy Trainin, and Wendy Smith. Developing k-8 computer science teachers' content knowledge, self-efficacy, and attitudes through evidence-based professional development. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1*, pages 540–546, 2022.

[6] Selcuk Dogan and Mete Akcaoglu. How teachers from underrepresented schools experienced a blended professional development program on computer science and game design. *Teacher Education Quarterly*, 51(1), 2024.

[7] Lee Allen. Computer science education: Online content modules and professional development for secondary teachers in west tennessee–a case study. *Journal of Educational Technology*, 15(2):8–15, 2018.

[8] Steve Tae Ho Kong. *A Case Study on Elementary Teachers' Experiences Teaching Computer Science*. PhD thesis, Concordia University (Oregon), 2019.

[9] Scott Sheridan, Chrystalla Mouza, and Lori Pollock. Professional development for computer science education: design and outcomes from a case study teacher. In *Society for Information Technology & Teacher Education International Conference*, pages 1115–1123. Association for the Advancement of Computing in Education (AACE), 2019.

[10] Selcuk Dogan, Rose Pringle, and Jennifer Mesa. The impacts of professional learning communities on science teachers' knowledge, practice and student learning: A review. *Professional development in education*, 42(4): 569–588, 2016.

[11] Lijun Ni. Building professional identity as computer science teachers: supporting secondary computer science teachers through reflection and community building. In *Proceedings of the Seventh International Workshop on Computing Education Research*, pages 143–144, 2011.

[12] Timothy Yuen, Maria Arreguin-Anderson, Guadalupe Carmona, and Matthew Gibson. A culturally relevant pedagogical approach to computer science education to increase participation of underrepresented populations. In *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*, pages 147–153, 2016. doi: 10.1109/LaTiCE.2016.44.

[13] SF Lam. Project-based learning. *Encyclopedia of the Sciences of Learning*, 2012.

[14] Khalid Mahmood Malik and Meina Zhu. Do project-based learning, hands-on activities, and flipped teaching enhance student's learning of introductory theoretical computing classes? *Education and information technologies*, 28(3):3581–3604, 2023.

[15] Texas Education Agency. Computer science 8–12 (241) preparation materials: Section 3: Overview and exam framework, 2024. URL https://www.tx.nesinc.com/Content/StudyGuide/TX_SG_obj_241.htm. Accessed: 20-Jul-2024.

[16] Michael Kölling. Using bluej to introduce programming. In *Reflections on the Teaching of Programming*, volume 4821 of *Lecture Notes in Computer Science*, pages 98–115. Springer, Berlin, Heidelberg, 2008. URL https://doi.org/10.1007/978-3-540-77934-6_9.

[17] Arduino. Arduino, 2024. URL https://www.arduino.cc/. Accessed: July 13, 2024.

[18] Microbit. Micro:bit, 2024. URL https://microbit.org/. Accessed: July 13, 2024.

[19] Lego Education Prime Project. Lego education spike, 2024. URL https://spike.legoeducation.com/prime/project. Accessed: July 13, 2024.

[20] Educative. C sharp vs java: Side-by-side comparison, 2023. URL https://www.educative.io/blog/c-sharp-vs-java. Accessed March 24, 2025.

[21] David J. Barnes and Michael Kölling. *Objects First with Java: A Practical Introduction using BlueJ*. Pearson, sixth edition, 2016. ISBN 978-013-447736-7, 978-1-292-15904-1.

[22] Youth Code Jam. Youth code jam, 2024. URL https://www.youthcodejam.org/about-us. Accessed: July 24, 2024.

[23] MIT Scratch. Scratch, 2024. URL https://scratch.mit.edu/. Accessed: July 13, 2024.

[24] BirdBrain Mission. Birdbrain technologies - about us, 2024. URL https://www.birdbraintechnologies.com/about-us/mission-history/. Accessed: July 13, 2024.

[25] BirdBrain Loan Program. Birdbrain technologies - loan program, 2024. URL https://www.birdbraintechnologies.com/loan-program/. Accessed: July 12, 2024.

[26] College Board. Ap computer science principles course, 2024. URL https://apcentral.collegeboard.org/courses/ap-computer-science-principles. Accessed: 20-Dec-2024.

[27] College Board. Ap computer science a course, 2024. URL https://apcentral.collegeboard.org/courses/ap-computer-science-a. Accessed: 20-Dec-2024.

[28] Texas Education Agency. Computer science 8–12 (241) preparation manual: Section 4: Sample selected-response questions, 2024. URL https://www.tx.nesinc.com/Content/StudyGuide/TX_SG_SRI_241.htm. Accessed: 20-Jul-2024.

[29] Chimi Dema and Ugyen Choden. Impact of project-based learning on computer science education. *Educational Innovation and Practice*, 7(1):29–54, 2024.

[30] App Inventor - About Us. Mit app inventor - about us, 2024. URL https://appinventor.mit.edu/explore/about-us.html. Accessed: 12-Jul-2024.

[31] App Inventor - Tutorials. Mit app inventor - our tutorials!, 2024. URL https://appinventor.mit.edu/explore/ai2/tutorials. Accessed: July 12, 2024.

[32] Congressional App Challenge. Congressional app challenge, 2024. URL https://www.congressionalappchallenge.us/. Accessed: July 13, 2024.

[33] Tinkercad. Tinkercad, 2024. URL https://www.tinkercad.com/circuits. Accessed: July 13, 2024.

[34] Unity. Unity real-time development platform, 2024. URL https://unity.com/. Accessed: July 13, 2024.

[35] National Center for Education Statistics. Acs school district profile 2018-22, 2024. URL https://nces.ed.gov/Programs/Edge/ACSDashboard/4838730. Accessed: July 12, 2024.

[36] Nina M Flores, Heather McBrien, Vivian Do, Mathew V Kiang, Jeffrey Schlegelmilch, and Joan A Casey. The 2021 texas power crisis: distribution, duration, and disparities. *J Expo Sci Environ Epidemiol*, 33(1):21–31, 2023. doi: 10.1038/s41370-022-00462-5. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC9851928/. Epub 2022 Aug 13.

[37] Texas Education Agency. Texas assessment program frequently asked questions, 2023. URL https://tea.texas.gov/student-assessment/student-assessment-overview/texas-assessment-program-faqs.pdf. Accessed: March 11, 2025.

[38] Texas Education Agency. 2017 accountability manual: Chapter 2 - ratings criteria and index targets, 2017. URL https://tea.texas.gov/texas-schools/accountability/academic-accountability/performance-reporting/2017manual-chapter2.pdf. Accessed February 13, 2025.

[39] Python. Welcome to python.org, 2025. URL https://www.python.org/. Accessed: March 24, 2025.

[40] Ivan Herrera and Patty Santos. San antonio isd proposes closing 19 schools as part of rightsizing plan, 2023. URL https://www.ksat.com/news/local/2023/09/19/san-antonio-isd-proposes-closing-19-schools-as-part-of-rightsizing-plan/. Accessed February 13, 2025.

[41] San Antonio Independent School District. Study on school building capacity data, 2023. URL https://www.saisd.net/page/study_sb_capacity_data. Accessed February 13, 2025.

[42] Isaac Windes. Saisd cuts positions as covid funding expires, 2024. URL https://sanantonioreport.org/saisd-cuts-federal-esser-covid-funding-expires/. Accessed July 18, 2024.

[43] Texas Instruments. Ti innovator(tm) rover, 2024. URL https://education.ti.com/en/activities/innovator. Accessed: July 16, 2024.

[44] Timothy T Yuen, Lucila D Ek, and Andrew Scheutze. Increasing participation from underrepresented minorities in stem through robotics clubs. In *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pages 24–28. IEEE, 2013.

[45] Ben Skudder and Andrew Luxton-Reilly. Worked examples in computer science. In *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148*, pages 59–64, 2014. URL https://crpit.scem.westernsydney.edu.au/confpapers/CRPITV148Skudder.pdf.