Building CS Teacher Capacity Through Comprehensive College/High School Partnerships

Robin Flatland
Siena College, New York, USA
flatland@siena.edu

James Matthews
Siena College, New York, USA
matthews@siena.edu

Pauline White
Siena College, New York, USA
pwhite@siena.edu

MaryAnne Egan
Siena College, New York, USA
maegan@siena.edu

Jesse Moya
Siena College, New York, USA
jmoya@siena.edu

ABSTRACT

Expanding access to and engaging diverse groups of students in high school computer science (CS) classes depends on qualified CS teachers. In this paper, we describe how faculty at our liberal arts college built CS teacher capacity at over 20 school districts through comprehensive college/high school partnerships. The majority of these districts serve rural or high-needs students, groups underrepresented in CS classrooms. The program works primarily with in-service teachers from other disciplines, helping them develop the expertise to teach CS. It is comprehensive in that it includes curricula and professional development for a high school level CS course and a dual-enrollment college level CS course, pathways to CS certification, community events, and opportunities for teacher leadership and collaboration. These modes of engagement are structured so that novice and veteran teachers and college faculty have opportunities to interact in different capacities over several years to create a robust professional learning community. Initial survey results show increasing levels of teacher confidence and sense of belonging, and increasing student confidence in their CS abilities.

CCS CONCEPTS

• Social and professional topics → Professional topics; **Computing education**; **K-12 education**;

KEYWORDS

professional development; dual-enrollment; teacher certification; professional learning community; broadening participation

ACM Reference Format:
Robin Flatland, James Matthews, Pauline White, MaryAnne Egan, and Jesse Moya. 2022. Building CS Teacher Capacity Through Comprehensive College/High School Partnerships. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022), March 3–5, 2022, Providence, RI, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3478431.3499364

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA

ACM ISBN 978-1-4503-9070-5/22/03…$15.00
https://doi.org/10.1145/3478431.3499364

1 INTRODUCTION AND RELATED WORK

With funding from a National Science Foundation grant [1], a team of faculty at our liberal arts college in upstate New York have built comprehensive college/high school partnerships that have increased computer science (CS) teacher capacity for over 20 school districts, and the number continues to grow. The majority of the high schools we are working with serve rural or high-needs students, groups that are underrepresented in CS classrooms [8]. Our program is comprehensive in that it provides a robust and complimentary set of professional development experiences that teachers can participate in to different degrees over many years, based on their level of experience and expertise. It is also sustainable due in part to several institutionalized college programs that are mutually beneficial to the college and the high schools.

Computer science is essential to a 21st century education [18], and yet less than half of all high schools in the US offer a CS course [8]. In the courses that are offered, female, rural, and minority students are underrepresented [8], depriving our society of the talents, creativity, and perspectives of diverse contributors. Expanding access to and successfully engaging diverse groups of students in high school CS classes depends on having qualified CS teachers. For many reasons, there is a national shortage of CS teachers. Schools have difficulty finding CS teachers due in part to flawed or non-existent state CS certification programs [14, 24], and high paying industry jobs draw people with CS skills away from careers in education. To illustrate the magnitude of the problem, in a two year period (2016 to 2017) less than 120 CS teachers graduated from universities with a CS degree compared to over 11,000 mathematics teachers [33].

A key strategy for addressing the shortage is to develop the CS teaching skills and expertise of in-service teachers from other disciplines. Professional development (PD) programs that build teacher capacity in this way typically focus on preparing teachers to teach a particular CS course. They often do this through one or more weeks of summer PD combined with regular shorter meetings and support during the school year, either on-line or in person. Two high profile examples include the Beauty and Joy of Computing (BJC) [5] and the Exploring CS (ECS) [10, 15] PD programs. The BJC PD prepares teachers to offer the BJC version [6] of the AP CS Principles course [2], and the ECS PD prepares teachers to offer the ECS high school level curriculum. For other examples see [4, 7, 17, 22, 23, 30]. A different approach used by the TEALS

[1]This work was supported by the National Science Foundation under Grant No. 1923378.

PD program [16] and others [12, 19] prepares teachers to offer a CS course by having them team teach it with a CS industry professional or college CS faculty.

The CS teacher shortage is of such a large scale that solving it will require a wide variety of initiatives over an extended period of time. In addition to initiatives like those above, we have created a model for building teacher capacity that also relies on high school teachers certified in other content areas. The model grew out of a research-practice partnership [29] between our college and local school districts with the aim of broadening participation through engaging high school CS classes. We see the training of in-service teachers as especially important for rural and high-needs schools that often struggle to hire and retain teachers.

Our model is multi-faceted and incorporates many strong features. First and foremost is extended PD experiences over several years led by college CS faculty. In our program, teachers can start with PD that prepares them to offer a high school level CS course. They can follow this with PD that prepares them to offer a dual-enrollment CS course which is a college-level course taught at the high school that students can take for both high school and college credit. In addition to content and pedagogy, the PD introduces equity issues in CS and strategies for recruiting/engaging underrepresented groups of students. These PD experiences can be extended further by taking up to 12 credit hours of college CS coursework which can lead to a second teaching certification in CS for our state, another feature of our model. A third feature is a robust professional learning community. In addition to the community built on shared PD and college coursework experiences, our model has college faculty and high school teachers participating together in community events such as local CS Teacher Association (CSTA) chapter meetings, high school programming contests, and professional presentations. This community helps eliminate feelings of isolation that many CS teachers experience as the only CS teacher in their school building [36].

Fletcher and Warner's CAPE Framework [13] is a useful tool for assessing equity in CS education. Our model addresses all four areas of this framework: Capacity, Access, Participation, and Experience. In this paper we will touch on all four aspects of the model, but the focus will be on building teacher capacity and increasing access to CS classes.

2 PARTICIPATING SCHOOLS

Prior to academic year (AY) 2019-2020, we were working with four high schools that offered our institution's introductory CS dual-enrollment course. During the past three years, our grant funded project has expanded this to 23 schools that are offering the dual-enrollment course or a newly created high school level Discovering CS course, or both. These schools serve rural, suburban, and city communities, with all but two located within 100 miles of our institution. Table 1a shows the number of participating schools by locale. The number of high-needs schools is shown in parentheses, where high-needs is defined as 50% or more students classified as economically disadvantaged (as reported by the NYS Education Department [28]). For example, Table 1a shows that this year (AY2021-2022) we are working with a total of 23 schools, 8 of them high-needs.

Three of the participating high schools are large (> 1000 students) high-needs city schools enrolling a combined total of 6235 students, with 62% of them Black or Latinx. We are also working with three large suburban high schools having a combined total of 4346 students, with 8% of them Black or Latinx and 19% coming from economically disadvantaged households. More than half of the high schools we are working with are rural with an average enrollment of 433. Four of the rural schools are also high-needs.

Table 1b shows the number of schools offering the high school level Discovering CS course and the dual-enrollment CS course (both described in Section 3.1), and the total number of students enrolled. (Note that some schools offered both courses.) The number of high-needs schools offering each course and the number of students from high-needs schools enrolled in each course are shown in parentheses. In AY2020-2021 for example, 8 schools offered Discovering CS, 3 of them high-needs; there were 211 students enrolled, of which 96 were from high-needs schools. (Student enrollment data for AY2021-2022 will not be available until the end of the school year.)

There are several ways we recruit new schools into our program. College faculty and high school teachers in our partnership regularly give talks on CS education at state mathematics educator conferences. The local CSTA chapter supports our institution's CS education work and informs their members about opportunities to work with us. Our reputation for providing quality PD and support is growing. This has resulted in several schools requesting to join our program after they heard about it from colleagues at other schools. As a means to address inequities in CS education, we have purposely sought to recruit high-needs and rural schools into our program. Part of our success has come from connections to our institution's math/education graduates. Five of the teachers we work with graduated from our institution. Three of these teachers participated in a NSF Noyce grant program as undergraduates that required a CS minor. These three teachers are working at high-needs schools, in part because it is a stipulation of the Noyce forgivable student loan program.

3 COMPONENTS OF THE COLLEGE/HIGH SCHOOL PARTNERSHIPS

Here we detail the components of our comprehensive college/high school partnerships for building CS teacher capacity: engaging curricula, summer and school year PD, pathways to CS certification, and community building events. These components are structured so that novice and veteran teachers and college faculty have opportunities to interact in different capacities over many years.

3.1 Engaging Curricula for Two CS Courses

We provide curricula, teaching materials, and professional development for two courses - a new high school level course called Discovering CS and a dual-enrollment college level course called Introduction to CS with Python and Multimedia. Both courses offer a broad introduction to CS and allow students to express themselves creatively through programming using images, sounds, and animations. The college faculty provide curricula, but we encourage and support teachers in adapting it for their particular contexts and sharing materials they develop with the group.

Table 1: (a) number of participating schools by locale, with number of high-needs schools in parentheses; (b) number of schools offering each course and students enrolled, with number of enrolled students from a high-needs school in parentheses.

(a)

	2019-2020	2020-2021	2021-2022
Rural	4 (0)	7 (1)	12 (4)
Suburban	6 (0)	7 (1)	7 (1)
City	3 (2)	4 (3)	4 (3)
Total schools	13 (2)	18 (5)	23 (8)

(b)

	2019-2020		2020-2021		2021-2022
	schools	students	schools	students	schools
Discovering CS	6 (2)	103 (62)	8 (3)	211 (96)	13 (5)
Dual-enrollment	7 (0)	102 (0)	14 (3)	144 (28)	14 (3)
Total Students		205 (62)		355 (124)	

When COVID-19 caused schools to go remote in 2020, the college faculty, high school teachers, and two undergraduate math/education majors collaboratively adapted the curricula for remote learning with NSF supplemental funding. To support students doing remote pair-programming on Chromebooks during the pandemic, they also adapted lab materials in both courses for browser based programming environments that support Google Docs style collaboration (i.e., Replit.com [32] and NetsBlox [25]). Sharing of instructional materials was done through curriculum maps with links. It was a challenging year, but it had the benefit of strengthening the college faculty/high school teacher relationships with teachers more involved in developing course materials.

3.1.1 *High School Level Course: Discovering CS.*

The high school level Discovering CS (DCS) course was developed to offer an opportunity for all students to feel welcome in a classroom where they could draw on their personal experiences and interests to grow their CS skills. It is a full year, collaborative lab and project focused course that dedicates an equal amount of time in three programming areas, as well as the impacts of computing. Throughout the course, students document and reflect on their course work on a Google site, which is shared with their teacher.

Although not required as a prerequisite, the DCS course is aligned with the dual-enrollment course and serves as a strong introduction to that course. The DCS course also affords schools the opportunity to align with our state's 9-12 CS learning standards that go into effect for all students in 2024 [26]. Some school districts offer the DCS course as a third-year math credit toward graduation requirements.

In the DCS course, students learn programming fundamentals in a blocks-based environment, through HTML/CSS/JavaScript, and again in text-based Python programming. Each of these programming experiences includes scaffolded laboratory experiences using pair programming. "Bells and whistles" are provided at the end of each lab to challenge students to take their learning deeper and/or in a creative direction. Pre-laboratory introductions make use of unplugged activities [3], related resources of interest such as video clips and games, guided notes, and Quizizz [31] to engage students and assess their understanding. Each programming experience culminates in an open-ended project allowing students the choice and creative freedom to demonstrate what they have learned.

Impacts of computing are explored through bi-weekly CS in the News discussions and activities that feature topics such as assistive technology & accessibility, art & architecture, algorithmic bias, and AI & ethics. Additionally, there are two week long projects. The first project focuses on issues of diversity and inclusion in CS. The second project focuses on issues of privacy, laws, and regulations related to computing. Throughout DCS, and especially through the impacts of computing, students gain an appreciation for the usefulness of CS in a wide variety of disciplines, as well as learning about the important positive contributions people of all backgrounds have and can make through computing.

3.1.2 *Dual-Enrollment Course: Introduction to CS with Python and Multimedia.*

The college level dual-enrollment course is a broad-based introduction to CS that includes multi-media programming with Python. It is similar in content and goals to the AP CS Principles course [2]. In fact, with some minor changes, one school had the curriculum approved for AP CS Principles. The dual-enrollment structure, however, offers several major advantages. One advantage is that students are assessed throughout the school year, with labs, homework, programming projects, and exams. The final course grade incorporates this wide range of summative assessments, rather than simply being based on one exam and a performance task, as is the case with AP CS Principles.

Following the Dale and Lewis CS Illuminated textbook [9], the course covers a breadth of topics: data representation, circuits and truth tables, hardware, and the von Neumann architecture, which serves as an introduction for assembly and machine language. It also introduces concepts from operating systems and artificial intelligence, and addresses societal impact. The programming language used is Python which is taught with a focus on multimedia applications using the Guzdial and Ericson textbook [21]. The students apply programming concepts by exploring and manipulating sounds and images, satisfying the desire of students to build things and see concrete results quickly [20].

The course is offered over a full year at high schools. Students can take it for high school credit only, or for both high school and college credit. As a prerequisite, we recommend that students complete two years of high school math and have the academic maturity to handle college level work. The cost for the 3 credits is $200 - this reduced price is similar to what other colleges charge for dual-enrollment credits. Our college waives the fee if a student qualifies for free or reduced lunch.

Another important advantage of a dual-enrollment course is that it sustains the relationships forged between the high school teachers and the college faculty beyond the initial first year of PD needed to start the course. Our college recently institutionalized faculty dual-enrollment coordinator positions. This position comes with release time so that every year the coordinator(s) can work with the teachers to provide new assessment materials and curriculum updates, respond to software and hardware issues, and ensure the quality of the program. Our institution justifies the cost of this position based on the course's tuition revenue and on the institution's increased name recognition at the participating schools. With many colleges experiencing declining overall enrollments [11] this name recognition is helpful with the recruitment of new students.

3.2 Summer and School Year PD

Since most of the teachers we are working with have little or no formal academic background in computer science, professional development is critical to our success in building CS teacher capacity. Of the 27 teachers we worked with, 22 are certified in math, 3 in technology, and 2 in English/Reading. Those who have never taught a CS class begin by teaching our Discovering CS course. As they gain experience and confidence, they can move on to the dual-enrollment course. Of the 6 teachers who offered Discovering CS in AY2019-2020, 4 of them went on to offer the dual-enrollment course in AY2020-2021.

All teachers offering a new CS course with us participate in a week of PD and planning before classes start. The PD is led by the college CS faculty, the same faculty that developed the curricula for the two courses and teach the second certification CS courses (see Section 3.3). We run PD for both the Discovering CS and the dual-enrollment course during the same week in parallel sessions. These sessions cover CS concepts, skills, and pedagogy used in the courses. The groups come together for lunches and for sessions on topics relevant to everyone. A highly valued part of the week is a session where our veteran teachers share their experiences teaching CS for the first time and give advice to the new teachers.

Sharing statistics, testimonies, research, and strategies about diversity and equity is also a featured part of this week. This culminates with teachers developing a plan for recruiting more females and minorities into their CS courses. During the school year they implement their plans, with opportunities to discuss their progress and challenges at follow up sessions on equity.

Additional PD occurs during the academic year. For the Discovering CS course, hourly PD meetings are held bi-monthly, along with two half-day Saturday sessions. These PD meetings provide teachers an opportunity to talk about the pace of the course, successes they have had with their students, and challenges they are facing. Teachers are encouraged to share ideas for overcoming challenges, as well as resources they have found useful and supplemental curriculum documents they have created. PD time is also used to review upcoming curriculum topics and to answer teacher questions about the curriculum. Saturday sessions are used to jump start the web development and Python programming curricular units. For the dual-enrollment course, teachers meet once per quarter to review upcoming content and to discuss the quarterly exams. Individual support sessions are held as needed and email exchanges are used to answer teacher questions.

3.3 Pathways to CS Certification

In 2018 the New York State Education Department (NYSED) established a CS teaching certificate. In 2019, our institution was the first to have a pre-service teacher CS certification program approved by the NYSED. We also provide a pathway for in-service teachers to qualify for a second teaching certificate in CS. For a second certificate in CS, NYS teachers need to complete 12 credit hours of CS coursework covering certain concept areas [27]. To facilitate this, we offer a selection of undergraduate CS courses at our institution in the late afternoons and summers, taught by CS faculty on this project. The grant provides funding to cover the tuition costs and fees for teachers.

To date we have offered three courses at our institution in this way: the dual-enrollment course Introduction to CS with Python and Multimedia (3 credits), a course on Java programming (4 credits), and a course on web design (3 credits). The number of teachers interested in taking courses and pursuing a second certification area exceeded our expectations. From Fall 2019 through Spring 2021, 14 teachers successfully completed 23 courses (79 credits), and we have eight teachers registered for courses in Fall 2021.

This program has benefits beyond the second certifications that teachers can earn. First, we encourage teachers planning to offer the dual-enrollment course to take it for credit at our institution as preparation for offering the course. Of the 17 teachers that have taught the dual-enrollment course, 10 of them took (or are currently taking) the course for credit at our institution. Second, taking courses is another way relationships are built among the enrolled high school teachers and the college faculty, thus contributing to the professional learning community.

Our institution has made a commitment to institutionalizing support for teachers taking CS courses after the grant funding ends. Specifically, they have agreed to waive tuition and all other fees for participating teachers. The advantage of offering waivers is that it supports the CS dual-enrollment program and raises the institution's profile with high school teachers. College administrators, especially those in enrollment management, see the advantages of partnering with schools on dual-enrollment course offerings.

3.4 Additional Community Building Events

As part of our goal to develop a robust community of practice around CS, we support other activities that members in our community participate in to different degrees and at different levels. For over 30 years we have held a high school programming contest. To encourage teachers and their students to participate, two weeks of class time in the dual-enrollment course is available for solving programming contest practice problems in teams. In the spring of 2020, before canceling the contest due to COVID-19, we had 72 teams (of 4 students each) registered for the contest from 18 high schools, with 8 being schools from this grant project. In 2021 we ran an on-line version of the contest with 61 teams from 20 schools, with 7 being schools from this grant project.

We also support our local CSTA chapter. The chapter meetings, when in-person, are held on the college campus, allowing us to arrange teacher PD sessions immediately afterwards. A CSTA meeting is also held annually during the high school programming contest. Combining CSTA meetings with PD sessions and the programming contest is not only convenient, but it also encourages teachers to become involved in CSTA. Our grant supports attendance at CSTA conferences, and teachers have taken advantage of this.

Connections between the college faculty and high school teachers are also strengthened through joint professional presentations. We have done several conference presentations together on CS education at mathematics educator conferences or CSTA conferences. These presentations also get new schools/teachers interested in working with us to start CS classes.

4 OUTCOMES

4.1 Teacher Outcomes

Each year we conduct pre- and post-surveys of the teachers to measure changes in their level of confidence in teaching the CS content and their sense of belonging to a CS learning community, which have been connected to improved student outcomes [1, 35]. Although our teacher sample sizes are not large, results are positive. They show that teacher efficacy and sense of belonging increases over time with participation in our program.

For teachers offering the dual-enrollment course, the average comfort level with the course content increased over time. As a representative sample, Figure 1 (left) shows the teachers' average responses when asked about their comfort level with teaching four topics from the course: data representation, circuits, artificial intelligence, and Python programming. The teachers responded using a Likert scale: 6-very comfortable; 5-comfortable; 4-somewhat comfortable; 3-not comfortable; 2-know almost nothing about this; 1- have no idea what this is. The Pre results (N=9) show the teachers' average response before offering the course for the first time. The Post 1 year (N=9) and Post 2+ years (N=7) results show the average response after teaching the course once and two or more times, respectively. Note that for three of the four questions, after 2+ years the average response was 6.0, the highest level of comfort.

For the Discovering CS course, the increases in the first year are more pronounced, likely because many of these teachers have no prior experience teaching CS. As a representative sample, Figure 1 (right) shows the teachers' average responses when asked about their level of comfort teaching four topics: Snap! conditionals, Snap! loops, Snap! custom blocks, and formal code tracing. The Likert scale used was the same 1 to 6 scale described earlier; the N values are 10 (Pre) 10 (Post 1 year), and 4 (Post 2+ years).

Similar positive trends over time are evident in survey questions related to the teachers' sense of belonging to a CS learning community. Teachers were asked to rate their level of agreement with statements such as those in Table 2 using a Likert scale: 7-strongly agree; 6-agree; 5-somewhat agree; 4-neither agree or disagree; 3-somewhat disagree; 2-disagree; 1-strongly disagree. The average responses are shown for teachers prior to starting our program (Pre-Mean), after one year (Post-Mean 1 year), and after two or more years (Post-Mean 2+). The most significant improvement was in opportunities to connect with other high school CS teachers which went up from 3.5 (before starting our program) to 6.7 (after 2+ years in our program).

In terms of retention, nearly all teachers have continued teaching CS into the second and third year of the program. Only two teachers have left the program. They were both part of the original group of 4 experienced teachers that started working with us prior to AY2019-2020. One of these teachers retired and the other was promoted to a district technology coach position.

4.2 Preliminary Student Outcomes

To measure how our work with teachers is impacting students, each year we conduct student pre- and post-surveys to assess their perceptions of their CS knowledge and level of interest in computer science. All questions are on a 1 to 5 Likert scale: 5-strongly agree; 4-agree; 3-neither agree nor disagree; 2-disagree; 1-strongly disagree. We present some preliminary findings from the surveys here, leaving a more thorough analysis for future work.

In an analysis of all student surveys from AY2019-2020 and AY2020-AY2021, 7 survey questions showed statistically significant changes from the beginning of the year to the end. See Table 3. Changes were determined to be statistically significant when the 95% confidence intervals of the pre- and post-means did not overlap. In each case the change was positive, with the biggest changes related to students' increased confidence in their understanding of CS concepts and programming skills (see first three questions). In the table, the columns show the average pre- and post-responses and their differences. When this analysis is performed on the survey results from each year separately, the results are consistent despite much of the instruction in AY2020-2021 being remote due to the COVID-19 pandemic.

Although we are focusing on building CS teacher capacity in this paper, we report some initial demographic and grade data on the students enrolled in the CS classes. Not all districts reported this data to us, but with data for 405 of the 560 CS students enrolled during years AY2019-2020 and AY2020-AY2021, Table 4 shows demographic results by locale. The right three columns show the number of students enrolled in a CS course, the percentage of these CS students that were Black/Latinx, and the average overall percent of Black/Latinx students attending the reporting schools. We observe that the percent of Black/Latinx CS students is close to the average overall percent of Black/Latinx students attending the schools in each locale. The percentage of females enrolled in a CS course was unchanged at approximately 25% in years AY2019-2020 and AY2020-2021. Given our PD work with teachers on increasing participation of underrepresented groups, we are disappointed that female enrollment did not increase in the second year. As we move forward, we have plans for working with school administrators and guidance counselors to address this. Grade data reported by districts shows 93% of students received a passing grade in their CS course. For female and Black/Latinx students, the pass rates were 96% and 86%, respectively.

5 CONCLUSIONS AND LESSONS LEARNED

Given the robust link between student learning and teacher capacity [34, 35], it is imperative to identify and develop better models for building teacher capacity. There is evidence in both teacher education [35] and CS education research [15] that providing sustained professional learning communities is key to promoting teacher capacity. Building teacher capacity involves fostering teachers' skills and motivation, as well as providing structures to support their success over time [34].

In this paper, we have described our work providing a comprehensive model for building teacher capacity that includes pathways to certification, job-embedded professional development (workshops and on-going support for curricular implementation), and opportunities for teacher leadership and collaboration. These various modes of engagement are structured so that novice and veteran teachers and college faculty members have opportunities to interact with each other in different capacities over several years. The core group of college faculty on the project instruct the college courses

Figure 1: Teacher comfort level with teaching the dual-enrollment (left) and Discovering CS course content (right).

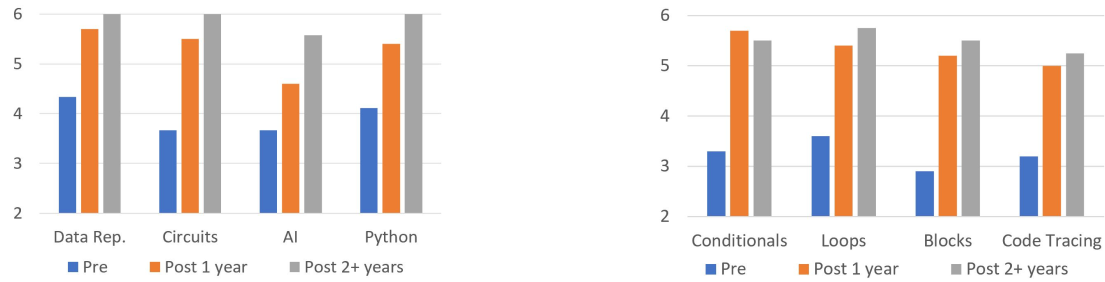

Table 2: Representative teacher survey questions with average pre, post 1 year, and post 2+ years responses.

Rate level of agreement using scale 1 (strongly disagree) to 7 (strongly agree)	Pre-Mean (N=14)	Post-Mean 1 year (N=15)	Post-Mean 2+ years (N=12)
If I have questions related to a computer science course I am teaching, there are college faculty who I feel comfortable reaching out to for answers.	6.2	6.7	6.9
I regularly have opportunities to talk with other computer science educators from other high schools to discuss issues related to teaching computer science.	3.5	5.0	6.7
I regularly have opportunities to talk with other computer science educators at CSTA meetings to discuss issues related to teaching computer science.	4.2	5.4	6.3
I belong to a professional learning community of computer science educators.	5.1	5.9	6.7

Table 3: Student survey questions with statistically significant changes from the pre- to post-survey.

Rate level of agreement using scale 1 (strongly disagree) to 5 (strongly agree)	Pre-Mean (N=624)	Post-Mean (N=493)	Difference
I am able to solve basic problems through computer programming.	3.02	3.92	0.90
I can explain what computer science is to my friends.	3.26	3.99	0.73
I have a basic understanding of the discipline of computer science.	3.44	3.98	0.54
I am a sophisticated computer user.	3.48	3.68	0.20
I am aware of career opportunities in computer science.	4.16	4.37	0.20
I have role models in computer science who are women.	2.75	2.96	0.20
I see examples of how computer science applies to my everyday life.	3.86	4.05	0.18

Table 4: Percentage Black/Latinx students by locale.

	CS Students	Black/Latinx CS Students	Overall School Black/Latinx
Rural	38	3%	4%
Suburban	204	6%	9%
City	163	62%	67%

necessary for CS certification, provide summer and school-year PD for course support, coordinate the programming contest and help to organize CSTA activities. This longitudinal professional learning community has helped to improve teacher confidence in teaching CS and enhanced their feelings of belonging to a CS community. In addition to this initial evidence of teacher capacity development, we have presented data that shows increases in students' confidence in their understanding of CS.

While the focus of the paper has been teacher capacity building, we have done this capacity work with an understanding of how it impacts other equity concerns like access, participation and the experiences of underrepresented students [13]. We intentionally reached out to rural and high-needs schools that are less likely to offer CS courses. We also made efforts to improve participation by providing teachers with strategies for recruiting students from underrepresented groups to computing. In addition, we provided curricula and instructional approaches intended to foster more equitable experiences. While the project has made positive strides towards promoting equity, we also realize there is considerable work to be done. For instance, the students participating in the CS courses are still overwhelmingly male. In addition, Black and Latinx students are somewhat underrepresented at some of our schools. We have recognized the importance of administrators and guidance counselors sharing responsibility with teachers for broadening participation at the school level. This should not be the sole responsibility of teachers.

The comprehensive model we have presented can be replicated at other higher education institutions looking to build collaborative partnerships with secondary schools. Dual-enrollment courses provide an initial structure for mutually beneficial relationships between colleges and high schools. However, we believe that the strength of the partnerships come from the various ways the project fosters building teacher capacity within a professional learning community providing on-going support for novice to veteran teachers.

REFERENCES

[1] Althauser, K. Job-embedded professional development: Its impact on teacher self-efficacy and student performance. *Teacher Development 19*, 2 (2015), 210–225.

[2] apcentral.collegeboard.org. AP Computer Science Principles. apcentral.collegeboard.org/courses/ap-computer-science-principles. Accessed: 2021-08-10.

[3] Bell, T., Witten, I., and Fellows, M. CS unplugged: An enrichment and extension programme for primary-aged students. https://classic.csunplugged.org/books/, 2015. Accessed: 2021-08-09.

[4] Beth, B., Lin, C., and Veletsianos, G. Training a diverse computer science teacher population. *ACM Inroads 6*, 4 (2015), 94–97.

[5] bjc.berkeley.edu. 2021 Professional Development: One-week workshop for teachers. bjc.berkeley.edu/summer-pd. Accessed: 2021-07-30.

[6] bjc.berkeley.edu. Beauty and joy of computing. An AP CS Principles curriculum. bjc.berkeley.edu. Accessed: 2021-08-10.

[7] Catete, V., Alvarez, L., Isvik, A., Milliken, A., Hill, M., and Barnes, T. Aligning theory and practice in teacher professional development for computer science. In *Proceedings of the 20th Koli Calling International Conference on Computing Education Research* (2020), pp. 1–11.

[8] Code.org. 2020 State of computer science education: Illuminating disparities, 2020.

[9] Dale, N., and Lewis, J. *Computer Science Illuminated*. Jones & Bartlett Publishers, 2019.

[10] exploringcs.org. ECS professional development. www.exploringcs.org/for-teachers-districts/teacher-support. Accessed: 2021-08-10.

[11] Fain, P. Enrollment declines steepest in midwest and northeast. Inside Higher Education. www.insidehighered.com/quicktakes/2018/05/22/enrollment-declines-steepest-midwest-and-northeast, 2018. Accessed: 2021-08-10.

[12] Flatland, R., Goldstein, I., Egan, M. L., Vandenberg, S., Fryling, M., and Small, S. Connecting colleges/universities and local high schools: A new model for high school CS teacher development. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE)* (2018), ACM, pp. 958–963.

[13] Fletcher, C. L., and Warner, J. R. Cape: A framework for assessing equity throughout the computer science education ecosystem. *Comm. ACM 64*, 2 (February 2021), 23–25.

[14] Gal-Ezer, J., and Stephenson, C. Computer science teacher preparation is critical. *ACM Inroads 1*, 1 (2010), 61–66.

[15] Goode, J., Margolis, J., and Chapman, G. Curriculum is not enough: the educational theory and research foundation of the Exploring Computer Science professional development model. In *Proceedings of the 45nd ACM Technical Symposium on Computer Science Education (SIGCSE)* (2014), ACM, p. 493–498.

[16] Granor, N., DeLyser, L. A., and Wang, K. TEALS: Teacher professional development using industry volunteers. In *Proceedings of the 47nd ACM Technical Symposium on Computer Science Education (SIGCSE)* (2016), ACM, pp. 60–65.

[17] Gray, J., Haynie, K., Packman, S., Boehm, M., Crawford, C., and Muralidhar, D. A mid-project report on a statewide professional development model for CS principles. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE)* (March 2015), ACM, pp. 380–385.

[18] Grover, S., and Pea, R. Computational thinking: A competency whose time has come. In *Computer Science Education: Perspectives on Teaching and Learning in Schools* (2017), Bloomsbury Academic, London, UK, pp. 19–38.

[19] Guzdial, M. Learning how to prepare computer science high school teachers. *IEEE Computer 44*, 10 (2011), 95–97.

[20] Guzdial, M. Balancing teaching CS efficiently with motivating students. *Communications of the ACM 60*, 6 (2017), 10–11.

[21] Guzdial, M., and Ericson, B. *Introduction to Computing and Programming in PYTHON*. Pearson Education, 2015.

[22] Guzdial, M., Ericson, B., McKlin, T., and Engelman, S. Georgia computes! An intervention in a us state, with formal and informal education in a policy context. *ACM Transactions on Computing Education 14*, 2 (2014), 1–29.

[23] Hamlen, K., Sridhar, N., Bievenue, L., Jackson, D. K., and Lalwani, A. Effects of teacher training in a computer science principles curriculum on teacher and student skills, confidence, and beliefs. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE)* (2018), ACM, pp. 741–746.

[24] Lang, K., Galanos, R., Goode, J., Seehorn, D., Trees, F., Phillips, P., and Stephenson, C. Bugs in the system: Computer science teacher certification in the U.S. A report by The Computer Science Teachers Association and The Association for Computing Machinery https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/CSTA_BugsInTheSystem.pdf, 2013. Accessed: 2017-06-16.

[25] netsblox.org. Netsblox: Innovative learning. netsblox.org. Accessed: 2021-07-26.

[26] nysed.gov. Computer science and digital fluency learning standards. www.nysed.gov/curriculum-instruction/computer-science-and-digital-fluency-learning-standards. Accessed: 2021-08-10.

[27] nysed.gov. Computer science certificate coursework guidance. http://www.nysed.gov/college-university-evaluation/computer-science-certificate-coursework-guidance. Accessed: 2021-07-26.

[28] nysed.gov. New York State education at a glance. data.nysed.gov. Accessed: 2021-07-26.

[29] Penuel, W. R., and Gallagher, D. J. *Creating Research-Practice Partnerships in Education*. Harvard Education Press, 2017.

[30] Qazi, M. A., Gray, J., Shannon, D. M., Russell, M., and Thomas, M. A state-wide effort to provide access to authentic computer science education to underrepresented populations. In *Proceedings of the ACM Technical Symposium on Computer Science Education (SIGCSE)* (March 2020), ACM, pp. 839–844.

[31] Quizizz.com. Quizizz: Engage everyone, everywhere. quizizz.com/teachers. Accessed: 2021-07-26.

[32] replit.com. Replit: Code, create, and learn together. replit.com. Accessed: 2021-07-26.

[33] Shein, E. The CS teacher shortage. *Communications of the ACM 62*, 10 (2019), 17–18.

[34] Stoll, L., Bolam, R., McMahon, A., Wallace, M., and Thomas, S. Professional learning communities: A review of the literature. *Journal of Educational Change 7*, 4 (November 2006), 221–258.

[35] Wei, R., Darling-Hammond, L., Andree, A., Richardson, N., and Orphanos, S. *Professional learning in the learning profession: A status report on teacher development in the United States and abroad*. National Staff Development Council, 2009.

[36] Yadav, A., S. Gretter, S. H., and Sandsl, P. Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education 26*, 4 (2017), 1–20.