International Journal of Computer Science Education in Schools, September 2022, Vol. 5, No. 4
ISSN 2513-8359

Computer Science Teacher Capacity: The Need for Expanded Understanding

David AMIEL[1]
Cynthia BLITZ[1]

[1] Rutgers University Center for Effective School Practices, New Brunswick, USA

DOI: 10.21585/ijcses.v5i4.151

Abstract

With the increasing need for the incorporation of computer science (CS) concepts into elementary and secondary education, it is imperative that the teaching workforce is adequately prepared to ensure that instruction in CS is robust, relevant, and aligned with appropriate learning standards, where appropriate. This paper shares results from a recent survey administered to current computer science educators across the K-12 space in the state of New Jersey. Using these results and recent literature, this research distills actionable, assessed needs to guide the provision of professional learning to ensure that educators have the necessary tools and knowledge to ensure robust and equitable implementation of computer science education. Results point towards a need to expand the present understanding of computer science by effectively differentiating CS from technology-based instruction and addressing an overrepresentation of analytical content domains, reaffirm a commitment to equity by acknowledging the persistent gaps in participation of marginalized student groups, and critically examine when and where the use of technology is necessary in delivering CS instruction.

Keywords: computer science education, elementary education, secondary education, professional learning, diversity equity and inclusion

1. Introduction

As a relatively new field, computer science (CS), and the education of it, are continuously expanding to meet growing economic, social, and cultural needs surrounding students' computational thinking and overall digital literacy (Webb et al., 2017). As the climate surrounding computing changes, society relies on the next generation of students to become competitive innovators and cultural drivers in a reality that is increasingly dependent on technology (Webb et al., 2017). At the same time, computer science education benefits students by challenging the way they think, and by teaching them to approach problems in novel and rigorous ways that are innately rooted in logical reasoning (Nager & Atkinson, 2016). According to the U.S. Bureau of Labor Statistics, jobs in the field of computer and information research are projected to increase by 22% by 2030, faster than the average for all other occupations (Bureau of Labor Statistics, 2022). Additionally, wages have been steadily increasing in the field, making computer science education an excellent opportunity for economic advancement (Nager & Atkinson, 2016).

Despite the value of computer science education (CSE), a recent study revealed that only 19 percent of high school seniors had taken any computer science courses (Nager & Atkinson, 2016). The study also a noted gender gap with only 14% of females having taken a computer science course, compared to 24% of males (Nager & Atkinson, 2016); female students of color are represented even less (Code.org et al., 2021). Researchers attribute this disparity to additional barriers faced by BIPOC learners, including lack of social encouragement, self-efficacy beliefs, academic exposure, and career perception (Nager & Atkinson, 2016). These participation gaps are also visible within the current computing workforce, in which women account for only 25% of positions across the world (Scott et al., 2017). Women of color are even more staunchly underrepresented in the workforce than their white counterparts, with African American, Latinx, Native American, Native Hawaiian, and Pacific Islander women in the U.S. comprising 20% of the population but only holding only 4% of positions in the computing workforce (Scott et al., 2017).

Even considering the growing centrality of CS in education, many current computer science teachers have never received adequate pre-service training in computational methods (Nager & Atkinson, 2016), meaning they may be unfamiliar with CS content knowledge and pedagogical practices. Many computer science teachers are also new to the field of education itself, with under 20% having more than 10 years of experience (Code.org et al., 2021). At this time, most states still do not require individuals teaching computer science courses to hold a certification in computer science (Code.org et al., 2021) and, at the time of writing, only 46% of computer science teachers across the country held a credential in computer science, with 23% of these teachers holding a CTE credential (Code.org et al., 2021), opposed to a credential dedicated to computer science.

Thus, pre-service teacher preparation and in-service professional development are promising avenues to both prepare teachers to deliver CS instruction and to introduce pedagogical practices that have shown promise in effectively conveying concepts in CS and computational thinking while broadening participation in the field, effectively tackling persistent participation and achievement gaps. At this time, many efforts have occurred at a national scale, including CSforAll and Code.org, among others.

1.1. The Case in New Jersey

Computer science has only recently become a standard discipline in public schools in both the U.S. and across the world (Andrew et al., 2016). Specifically in New Jersey (NJ), although computer science is not a graduation requirement, high schools are required to *offer* CS classes. Since this mandate was enacted in January 2018 (and going into effect that fall), school-level implementation of classes have been a challenge: in 2020, only 68% of high schools offered any form of computer science course (Code.org et al., 2021). Recently, NJ has adopted student learning standards in computer science and design thinking, which span the entire K-12 space.

For the first time, starting in the 2022 school year, CSE will become a mandatory part of education in the K-8 space in NJ schools. Similar trends are unfolding nationwide, with schools across the United States beginning to adopt computer science across the K-12 spectrum (Code.org et al., 2021). Organizations like *CS is Elementary* are working to spread awareness and increase adoption of computer science instruction at lower grade levels. Given that unlike other subject areas, computer science does not benefit from decades of research on best practices in content delivery and teacher preparation, efforts seeking to collect information about where computer science is being taught and what that instruction looks like will be important in understanding where and how CSE is successful.

Alongside such learning standards, educators will require support as they prepare to implement CSE in their classes. This professional learning, although inclusive of content and pedagogical knowledge, must also extend beyond the traditional scope of professional development to increase understanding of the broad scope and definition of the discipline itself. This is especially important considering that the New Jersey standards, as well those from the national Computer Science Teachers Association standards (CSTA), include elements that extend beyond coding and analysis, which are commonly misunderstood to represent the *entirety* of CS.

Additionally, the timely implementation of computer science standards provides a platform to address the persistent issues of equity and participation in computer science education. As all students (regardless of race, gender, socioeconomic status, or ethnicity) will be required to receive computer science instruction. This provides educators with opportunity to broaden the CS pipeline from younger grade levels and promote more authentic and responsive engagement with the subject.

To support this work, New Jersey has piloted three professional learning hubs centered at universities across the state. These hubs, based on the model of the United Kingdom's Teaching School Hubs (Department for Education), are designed to provide professional learning to educators across the state to equip them with the content and pedagogical knowledge (inclusive of culturally responsive and equitable practices) needed to ensure the success of the standards' implementation.

2. Methods

The purpose of this research is to generate a snapshot of computer science educators in New Jersey that can inform the allocation of resources for the intentional, effective provision of professional learning. To this end, a 20-question survey was developed and distributed to attendees of an annual computer science education summit. The research acknowledges a sampling bias of those completing the survey, who are significantly more likely to themselves be computer science teachers, or at least involved with CSE in some way. However, although the survey was administered to a convenience sample, given that the purpose of their insight is to inform resource allocation, we are confident that the assessed needs of this sample will provide, at the very least, a lower bound for the more broadly needed supports (as those completing the survey are likely to be among those receiving such supports). The survey contained several demographic questions, a series of scale items adapted from a similar instrument (Banilower et al., 2018), as well as items directly related to the implementation of NJ CS Standards. *Appendix A* contains the survey questions that were used in the analyses discussed in this paper. This research sought to address the following questions:

1. What are teachers' perceptions of their own knowledge of various content domains of computer science, and how confident are teachers in delivering CS instruction? (*Instructional Practices*)
2. What are teachers' perceptions of the barriers to implementing rigorous computer science instruction? (*Institutional Practices*)
3. How can professional learning better be tailored to bolster teacher capacity using findings related to teachers' perceptions of the above instructional and institutional practices in CSE?

All routine data analyses took place in SPSS Version 28, including the creation of scale item aggregate scores, the generation of frequency tables, and the calculation of reported means and standard deviations. This survey is part of a larger campaign to gather robust and relevant information on the landscape of computer science educators across the state and disseminate such information to educational agencies, policymakers and advocates, and professional learning providers to maximize the cumulative impact of efforts to support computer science education. This study received approval from the Rutgers University Institutional Review Board.

3. Results

3.1. About the Sample

The survey was sent to a total of 93 individuals and received a total of 41 responses; after removing abandoned submissions and those that did not pass screening questions, 29 valid responses remained and were used in the analysis discussed throughout this paper. After an initial response rate of 44%, 31% of responses were used in analysis. Of this sample, 83% (n=24) identified as female, and the remaining 17% (n=5) identified as male. The vast majority, 90%, of respondents identified as White or Caucasian, while 3.3% identified as Black or African American, 3.3% Asian American or Pacific Islander, and 3.3% as mixed race (Black & Caucasian). 10% reported that they have no formal training or education specifically in computer science, 35% reported formal training or education less than an undergraduate minor in CS, 7% reported obtaining an undergraduate CS minor, 38% reported an undergraduate CS major, and 10% reported their highest education in computer science is a graduate degree.

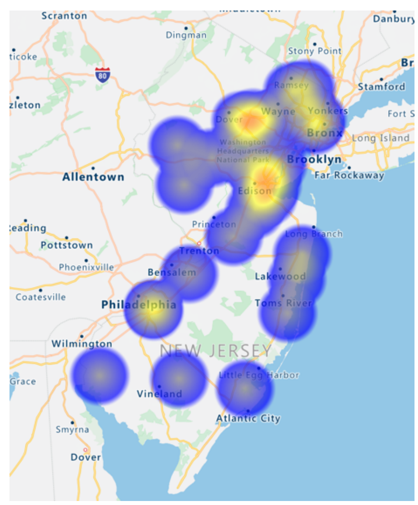

Figure 1. Heatmap of Analyzed Responses

Figure 1 shows a heatmap of valid survey responses received across the state, with yellow and orange indicating a higher concentration of responses. Each data point represents the geographic region of where the individual teaches computer science, not necessarily their place of residence. Of these, 52% of individuals reported teaching in a district where most students are from poor or working-class families, 41% reported teaching in a district where most students are from middle class families, and the remaining 7% reported teaching in predominately wealthy districts.

Most respondents indicated teaching computer science most recently at the high school level, with 33% indicating they teach computer science in grades 11-12 and 30% indicating they teach computer science in grades 9-10. Outside of high school grades, 17% indicated they most recently taught computer science in grades 6-8, 11% in grades 3-5, and 9% in grades PreK-2. Further, the majority (55%) indicated that they exclusively teach computer science, with 21% indicating that 0% - 50% of their course load is computer science and 24% reporting between 50% and 75%.

Table 1. Teachers' Perceptions of Content Knowledge of CS Standard Domains (N=29)

	I have not heard of these concepts (1)	**I have heard of these concepts (2)**	**I am familiar with these concepts (3)**	**I have mastered these concepts (4)**
Algorithms & Programming	0%	10%	45%	45%
Computing Systems	0%	7%	55%	38%
Data & Analysis	0%	17%	48%	35%
Effects of Computing	7%	17%	52%	24%
Engineering Design	3%	17%	59%	21%
Ethics of Computing	0%	24%	48%	28%
Impacts of Computing	0%	14%	45%	41%
Interaction of Humans & Machines	3%	7%	59%	31%
Nature of Technology	3%	28%	48%	21%
Networks & the Internet	0%	7%	66%	28%

3.2. Instructional Practices

Participating educators shared insights on their self-perceptions of knowledge for each content domain covered in the 2020 New Jersey CS Student Learning Standards. On a scale from 1 ("I have not heard of these concepts") through 4 ("I have mastered these concepts"), educators reported they were most comfortable with algorithms and programming, with an average weighted score of 3.34, with standard deviation of 0.67. From highest to lowest weighted score, educators rated their confidence with the various disciplinary domains as computing systems (3.31, SD=.60), impact of computing (3.28, SD=.70), networks and the internet (3.20, SD=.55), data and analysis (3.17, SD=.71), interaction of humans and machines (3.17, SD=.71), ethics of computing (3.03, SD=.73), engineering design (2.96, SD=.73), effects of computing (2.93, SD=.84), and nature of technology (2.86, SD=.79). *Table 1* further details the distribution of responses for each content area.

Participants also shared how prepared they feel in delivering computer science instruction. On a scale of 1 ("Not adequately prepared") to 4 ("Very prepared"), teachers rated, on average, that they were most prepared to teach the relevance of computer science (3.6 weighted score, SD=.56), followed by teaching introductory computer science (3.55, SD=.57), ensure equal student participation (3.55, SD=.69), garner student interest in computer science (3.52, SD=.51), foster group interactions with students (3.45, SD=.69), reach students from traditionally underrepresented populations (3.41, SD=.73), teach using a guided inquiry approach (3.14, SD=.74), and utilize culturally relevant teaching and associated pedagogies (2.93, 70); teachers reported they felt least prepared to teach computer science to English Language Learners (2.34, SD=.81). *Table 2* shows the distribution of responses related to participants' preparation for computer science instruction.

Table 2. Teacher-Reported Levels of Preparation to Deliver Computer Science Instruction (N=29)

	Not Adequately Prepared (1)	**Somewhat Prepared (2)**	**Moderately Prepared (3)**	**Very Prepared (4)**
Differentiate Instruction	0%	28%	35%	38%
Teach ELLs	10%	55%	24%	10%
Generate Student Interest	0%	0%	48%	52%
Reach Underrepresented Minorities	0%	14%	31%	55%
Teach Relevance of CS	0%	3%	31%	66%
Use Guided Inquiry	0%	21%	45%	35%
Ensure Equal Participation	0%	13%	24%	66%
Foster Group Interactions	0%	13%	35%	55%
Teach Introductory Concepts	0%	3%	38%	59%
Utilize CRT/Pedagogies	0%	28%	52%	21%

3.3. Institutional Practices

Participating educators shared their perceptions of challenges to implementing rigorous computer science education at the school level. On average, the greatest challenges shared was rapidly changing technology (with a weighted score of 1.97/3, SD=.68), closely followed by a lack of hardware and software resources (1.93, SD=.70). Other prominent challenges included difficult subject matter (1.76, SD=.64) and a lack of support from the school and staff (1.76, SD=.64). *Table 3* further details these responses.

Educators also reflected on student interest and broadening participation in computer science education at their institutions. At the high school level[1], less than 20% of respondents indicated that the demographic composition of their computer science classrooms was identical or nearly identical to the demographic composition of their school. Around 30% indicated that the two demographic groups were slightly different, and 56% indicated that they were moderately different. It is also worth noting that the participants that indicated a (nearly) identical match also reported they teach in districts with relatively homogeneous demographic composition. Further, participants shared that in the last 5 years, the demographic composition of the computer science classroom is becoming more like the demographic composition of the school. All educators responded that their classrooms were moving more towards equal demographic composition, with the most common response being a 7 on a scale of 1 ("much less similar") to 10 ("much more similar").

When considering student interest in computer science education, most teachers indicated that there is strong student interest at their school for computer science (with a combined score of 3.79/4, SD=.94), but "enrolling in computer science is a student priority" received the lowest responses in the category (2.93, SD=.75). Teachers indicated that students taking introductory computer science tend to move to more advanced classes (3.72, SD=.59), student dropout of computer science classes is low (3.41, SD=1.0), and that there is demand for more computer science classes (3.44, SD=.95).

4. Discussion

With the insights from the results discussed above, the research sought to distill actionable, assessed "needs" to inform the path forward in the provision of professional development to K-12 educators surrounding computer science education. These needs, situated with relevant literature, offer suggestions for resource allocation and better articulation of programming to maximize the impact of efforts by organizations in this space.

4.1. Encompassing Need: Expand the Present Understanding of Computer Science

The idea that computer science is synonymous with programming remains a common misconception in computer science education. This pervasive myth is true for many prospective computer science students, who often do not consider that while programming is indeed an important component, professionals in the field also engage in activities that include hardware and software design, research into the impact on society (Barr & Stephenson, 2011; Denning, 2004; Denning et al., 2017; Pardaboyevich et al., 2021) and routine engagement in problem solving (Peckham et al., 2000). Survey results show that the majority of computer science teachers also have a background in teaching mathematics (with a large number currently teaching both subjects), which could contribute to the overrepresentation of these analytical concepts. Further, teachers rated their understanding of content knowledge for Algorithms & Programming most highly out of all content domains, whereas Nature of Technology, Effects of Computing, and

Table 3. Teachers' Perceptions of Challenges to Implement CSE (N=29)

	Minor/no Challenge	**Moderate Challenge**	**Great Challenge**
Lack of Student Interest	48%	38%	14%
Rapidly Changing Technology	24%	55%	21%
Difficult Subject Matter	28%	59%	14%
Lack of School/Staff Support	35%	55%	10%
Lack of Student Knowledge	35%	55%	10%
Lack of Curriculum Resources	45%	38%	17%
Lack of Software/Hardware	28%	52%	21%

[1] To control for the fact that computer science instruction, when offered at the K-8 level, is often a mandatory elective, only responses that indicated teaching at the 9-12 level were considered for this question.

Ethics of Computing (all non-analytic) received the lowest ratings of confidence (Nature of Technology received an average rating of 2.86, which is over 1.5 deviations below average). These results are not unique to this survey; a recent study assessed the perceptions of digital technology teachers attending a computer science workshop in which a majority of teachers surveyed indicated that they felt that computer science is mostly about programming (Prieto-Rodriguez & Berretta, 2014).

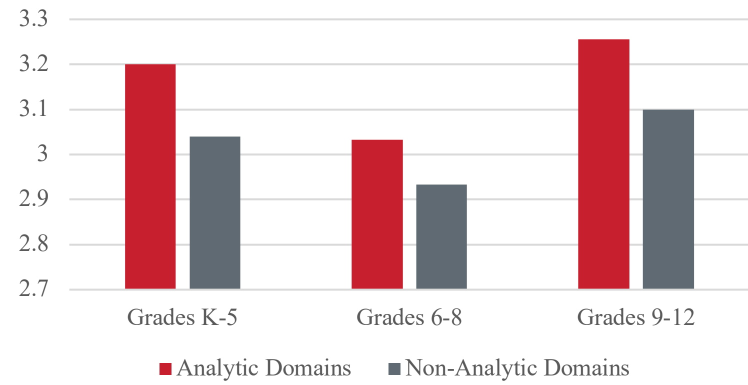

Figure 2. Self-Rating of Content Knowledge Across Grade Band & Analytic v. Non-Analytic Content Domains

As illustrated in *Figure 2*[2], teachers rated their content knowledge, on average, nearly a full standard deviation lower in non-analytic domains when compared to analytic ones. This difference is seen in all grade bands, though is most distinct in grades K-5. It is worth noting that teachers indicating they currently teach introduction to computer science rate their understanding of analytic concepts 0.26 points higher than non-analytic – a full deviation above the mean. This is not to say that these teachers do not understand non-analytic concepts but suggest that an overemphasis of more mathematical aspects of computer science can begin at the introductory level, as teachers may feel more comfortable in providing deeper insight, answering questions more completely, or offering more thoughtful discussion in those areas where they have a stronger understanding.

These findings present an urgent need to expand the current understanding of computer science as a discipline, both to students and to educators. Survey results, combined with analyses of current institutional practices, show that computer science is often contained in the mathematics department and that CS courses are classified as mathematics electives (Blitz et al., 2021); this can present computer science to students as a close alternative to math. The overemphasis of analytic and mathematical concepts in computer science education can not only deter students from participating in computer science but has also constructed a false barrier-to-entry for educators, especially in the K-8 spaces, who may be concerned about their own knowledge of the subject or not see the value in introducing computer science in younger grades (Barr & Stephenson, 2011; Denning, 2004; Denning et al., 2017). Professional learning providers therefore must challenge these conceptions of computer science and more broadly and intentionally disseminate information that highlights computer science as a multidisciplinary area of study centered around problem solving, computational thinking, and the creation of technology to meet today's demands.

4.2 Ancillary Need: Reaffirm Commitment to Equity

Expanding the present understanding of computer science in this way will be a fundamental part of bringing computer science to the K-8 space (and in New Jersey, in the implementation of upcoming standards), and it will also be a powerful tool in moving forward the work of ensuring equitable participation in computer science. It will be important for educators to authentically engage young learners with the full spectrum of computer science to encourage long-term participation. In this way, computer science can 'start broad' to invite engagement, rather than starting with narrow specialization such as programming or web design, as is often the case when computer science is first offered at the high school level.

Across the board, teachers felt strongly that they were able to teach the relevance of computer science to their students (with a combined score of 3.62/4, but an incomplete understanding of computer science may be hindering these efforts (especially when considered alongside respondents identifying "lack of student interest" as the most minor challenge to computer science education). Survey results show that over 40% of educators report that their computer science

---

[2] *Figure 2* shows the average scores received from 29 survey participants when asked to rate their level of mastery of content knowledge in the various domains of computer science and design thinking. Content domains were categorized as analytic (engineering design, algorithms and programming, data and analysis, networks and the internet, and computing systems) or non-analytic (nature of technology, effects of computing, ethics of computing, human-computer interaction, and impacts of computing), and ratings were averaged across each category.

classroom is at least moderately different in demographics from their overall student population; when considering only high school classes, where CS is most likely to be an 'opt-in' course, that number increases to 56%. However, results also show that educators feel strongly that there is student interest in computer science. These results point to a need to reaffirm educators' commitment to equity in CSE and broadening participation in computing; professional development should not only increase awareness of the persistent inequities in the field but equip educators with culturally relevant pedagogies in CS to tackle such disparities.

4.3. Ancillary Need: Equip Educators with Ways to Leverage Existing, Available Technology

Recent years have seen a sharp increase in the availability of "plug and play" curricula, which are implementation-ready programs to bring computing instruction to classrooms. To educators, the ongoing release of new curricular options using rapidly developing computer technologies, although to the same end, can be overwhelming (Pardaboyevich et al., 2021). In fact, rapidly changing technology was cited as the number one challenge to teaching computer science among both new and veteran teachers. Although most of these curricula teach to the same set of standards (those created by CSTA), many do so through different approaches. Some take an approach through physical computing, others through app development, others center around drag-and-drop programming, and so on.

As such, although rapidly changing technology is perceived as a challenge (one that was cited as at least a moderate challenge by around 75% of survey respondents), it may be a manufactured one. To respond, professional learning providers should clearly articulate that although there are many curricular and technological resources available, it is likely the case that only one, if any, will be needed to deliver quality CSE, even as new resources become available. A school's curriculum (whether purchased as a pre-created curriculum or developed) does not need to respond to each release of a new platform; similar research has shown that teachers require professional development that is aligned with their curricular needs (Qian et al., 2018), which does not always call for the use of new, or any, computing devices.

Following closely behind is the lack of software or hardware resources, which was cited as at least a moderate challenge by 73% of survey respondents. As work is done to frame computer science education as a discipline centered around problem solving and computational thinking in addition to programming, it will be necessary to understand that the newest instructional or curricular resources are not always necessary to implement and sustain a rigorous computer science education. Additionally, a promising way to engage students in computational thinking without the need for expensive equipment is through the use of "CS Unplugged" activities (Bell et al., 2009). Although these instructional practices have demonstrated efficacy across the board, they can be particularly helpful in under resourced schools (Bell et al., 2009).

5. Conclusion

As computer science is being recognized as a driving force in our world, a useful lens to understand and process the world around us, and a vehicle for financial growth, computer science education is becoming an increasing part of students' academic experience. This increase is facilitated by efforts such as the implementation of learning standards, increased resource allocation for clubs or extra-curricular activities, and national efforts like the CSforAll program or the nonprofit *CS is Elementary*. However, to ensure that these efforts are fully realizing their potential (and that all students are effectively reached and engaged in the process), it is necessary for educators to have a more complete understanding of computer science, which will increase the efficacy in the adoption of learning standards, promote equity in CSE and broadening participation in computing, and dismantle challenges perceived by CS educators. Professional learning providers, such as regional learning Hubs or curriculum developers, and other advocating agencies must act quickly to fill in gaps in understanding as educational interventions are implemented and scaled.

5.1. Implications for Future Work & Study Limitations

Results from the present study offer a promising direction for the provision of professional learning to educators in computer science education. As providers work to address the research-identified dearth of rigorous professional learning opportunities for computer science teachers (Yadav et al., 2016), it will be crucial to ensure that educators have a complete, nuanced understanding of what computer science is as a discipline. First, teachers' under-confidence in non-analytic content areas of computer science illustrate the need for an intentional emphasis on aspects of CS that extend beyond algorithms and computer programming, such as problem solving, computational thinking, and the social impacts and implications of a technology-driven world. In doing so, professional learning has the potential to address misconceptions about computer science as a discipline and tackle false barriers to entry caused by inaccurate equivalences between CS and mathematics. Further, in broadening the present understanding of computer science, educators, especially in elementary spaces and those where CS instruction is mandatory for all students, will be better equipped to cast a wider net while delivering CS instruction, effectively broadening the CS pipeline at its early entry-points. Additionally, building awareness of the breadth CS will help address teachers' primary concerns of rapidly changing technology and lack of hardware and software resources by better equipping educators to teach CS in ways that do not solely rely on (ever-changing) computing devices or canned curricula.

Although we make the case that an expanded teacher understanding of the complexities and full spectrum of computer science can – directly or indirectly – address many of the present challenges in CSE, we also acknowledge that our study is exploratory in nature and thus presents a set of limitations. First, we recognize that our survey sample is likely a biased one, as recruitment for survey respondents took place at a computer science education summit. However, given that our sample is likely more CS-inclined, we propose that the challenges distilled from these findings can serve as a lower bound towards generalization. Second, we caution readers to consider our proposed need for an expanded understanding as a backdrop for interpreting survey results.

Finally, as both a limitation of this study and a promising avenue for future work, results illustrate the need for a better understanding of the current CS teacher landscape. Input from various educational stakeholders is necessary in determining the best direction for professional learning, and although this study hopes to provide a solid foundation, we recognize our limited sample and call for future research to validate and expand findings discussed herein. We also acknowledge that this study, along with several points from the survey instrument, are narrowly focused on the current CSE ecosystem in New Jersey and may not be directly generalizable to other geographic regions, though we hope that our approach and findings may be important markers.

Acknowledgements

The authors would like to acknowledge the funding received from the National Science Foundation under grant award 1837305 through the CSforAll RPP Program, which supported, in part, the development and administration of the survey in the above research.

References

Andrew, F., Mary, W., Margaret, C., Charoula, A., Joyce, M.-S., Joke, V., & Jason, Z. (2016). Arguing for Computer Science in the School Curriculum. *Journal of Educational Technology & Society*, *19*(3), 38-46. http://www.jstor.org/stable/jeductechsoci.19.3.38

Banilower, E. R., Smith, P. S., Malzahn, K. A., Plumley, C. L., Gordon, E. M., & Hayes, M. L. (2018). *Report of the 2018 NSSME+*. I. Horizon Research.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer Science Unplugged: school students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*.

Blitz, C., Allen, V., & Amiel, D. (2021). *Recruiting Diverse Learners to High School Computer Science* Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, Virtual Event, USA. https://doi.org/10.1145/3408877.3439565

Bureau of Labor Statistics. (2022). Occupational Outlook Handbook, Computer and Information Research Scientists. In: U.S. Department of Labor.

Code.org, CSTA, & Alliance, E. (2021). *2021 State of computer science education: Accelerating action through advocacy*. https://advocacy.code.org/stateofcs

Denning, P. J. (2004). The field of programmers myth. *Commun. ACM*, *47*(7), 15–20. https://doi.org/10.1145/1005817.1005836

Denning, P. J., Tedre, M., & Yongpradit, P. (2017). Misconceptions about computer science. *Commun. ACM*, *60*(3), 31–33. https://doi.org/10.1145/3041047

Department for Education. *Teaching school hubs*. https://www.gov.uk/guidance/teaching-school-hubs

Nager, A., & Atkinson, R. D. (2016). The Case for Improving U.S. Computer Science Education. *SSRN Electronic Journal*. https://doi.org/https://doi.org/10.2139/ssrn.3066335

Pardaboyevich, R. F., Abdunazirovich, U. S., & Saydullayevich, S. Q. (2021). Teaching Computer Science at School - Current Challenges And Prospects. *JournalNX - A Multidisciplinary Peer Reviewed Journal*, *6*(11), 217-221. https://doi.org/https://repo.journalnx.com/index.php/nx/article/view/146

Peckham, J., DiPippo, L., Reynolds, J., Paris, J., Monte, P., & Constantinidis, P. (2000). A first course in computer science: the discipline is more than programming. *J. Comput. Sci. Coll.*, *15*(5), 220–227.

Prieto-Rodriguez, E., & Berretta, R. (2014, 22-25 Oct. 2014). Digital technology teachers' perceptions of computer science: It is not all about programming. 2014 IEEE Frontiers in Education Conference (FIE) Proceedings,

Qian, Y., Hambrusch, S., Yadav, A., & Gretter, S. (2018). Who Needs What: Recommendations for Designing Effective Online Professional Development for Computer Science Teachers. *Journal of Research on Technology in Education*, *50*(2), 164-181. https://doi.org/10.1080/15391523.2018.1433565

Scott, A., Martin, A., McAlear, F., & Koshy, S. (2017). Broadening participation in computing: examining experiences of girls of color. *ACM Inroads*, *8*(4), 48–52. https://doi.org/10.1145/3149921

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when? *Education and Information Technologies*, *22*(2), 445-468. https://doi.org/10.1007/s10639-016-9493-x

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education*, *26*(4), 235-254. https://doi.org/10.1080/08993408.2016.1257418

Appendix A

Selected Survey Questions

1. [Multiple Choice Matrix] How would you rate your level of content knowledge in each of the following areas related to computer science? (*Cols: I have heard of these concepts, I am familiar with these concepts, I understand these concepts, I have mastered these concepts*)

- Computing Systems
- Networks and the Internet
- Impacts of Computing
- Data & Analysis
- Algorithms & Programming
- Engineering Design
- Interaction of Technology and Humans
- Nature of Technology
- Effects of Technology on the Natural World
- Ethics & Culture

2. [Multiple Choice Matrix] How prepared do you feel to do each of the following in your computer science instruction? (*Cols: Not adequately prepared, Somewhat prepared, Moderately prepared, Very prepared)*

- Plan differentiated instruction for your students
- Teach computer science to English language learners
- Encourage students' interest in computer science
- Teach computer science to underrepresented student populations (females, racial or ethnic minorities, students who are economically disadvantaged)
- Teach students the relevance of computer science in their daily lives
- Teach computer science using a guided inquiry approach
- Ensure that every student in the class participates in the learning activities
- Foster group interactions during the learning activities
- Plan and facilitate learning activities focused on introductory computer science concepts
- Utilize culturally relevant pedagogical practices

3. [Multiple Choice Matrix] How would you rate each of the following potential challenges to teaching CS? (*Cols: Minor/No challenge, Moderate challenge, Great challenge*)

- Lack of student interest/enrollment
- Rapidly changing technology
- Difficult subject matter
- Lack of support/interest from school staff
- Lack of student subject knowledge
- Lack of curriculum resources
- Lack of hardware/software resources

4. [Multiple Choice Matrix] How much do you agree or disagree with the following statements? (*Cols: Strongly Disagree, Disagree, Agree, Strongly Agree*)

- There is strong interest from students in the computer science courses offered in my school.
- Enrolling in computer science courses is a top priority for students in my school.
- Students who enroll in introductory computer science courses typically want to take an advanced course.
- Student dropout from computer science courses offered in my school is low.
- There is student demand for more computer science courses in my school.

5. [Multiple Choice] How well does the current demographic composition of your computer science classroom match the demographic composition of your school?

- They are (nearly) identical
- They are slightly different
- They are moderately different
- They are significantly different
- They are dramatically different

6. [Sliding Scale] In the last 5 years, has your classroom demographic makeup become more or less similar to the demographic makeup of your school? (*Scale from 1-10, Labels: 1= Classroom has become much less similar, 5 = No change, 10 = Classroom has become much more similar*)