PRIMING THE COMPUTER SCIENCE TEACHER PUMP

Integrating Computer Science Education into Schools of Education

Leigh Ann Delyser, CSforALL
Joanna Goode, University of Oregon
Mark Guzdial, Georgia Institute of Technology
Yasmin Kafai, University of Pennsylvania
Aman Yadav, Michigan State University

Priming the Computer Science Teacher Pump: Integrating Computer Science Education Into Schools of Education

Authors:

Leigh Ann Delyser, CSforALL

Joanna Goode, University of Oregon

Mark Guzdial, Georgia Institute of Technology

Yasmin Kafai, University of Pennsylvania

Aman Yadav, Michigan State University

Acknowledgements

We would like to thank all of the organizations and individuals that made this report possible. The initial funding for the project was provided through grant #1636881 by the National Science Foundation. Any opinions, findings, or conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

We thank Microsoft for hosting both the workshop as well as the report launch event.

A heartfelt thank you to all of the individuals responsible for the website design and updates, editing, and reviewing of this report, including: Renu Grover, Gabi Rivkin, Leeba Erlbaum, Frances Schick, and Natalie Avalos.

We are especially grateful to our workshop attendees, who played a critical role in developing the insights and recommendations found within this report. Without your contribution, this report would not have been possible. The names of our workshop attendees are listed at the end of the report.


IONA COLLEGE
COMPUTER SCIENCE
THE QUEENS SCHOOL OF INQUIRY
EAT.CODE.SLEEP.REPEAT.
2016 - 2017
UG DEALER
CS FAIR
Student
csnyc

TABLE OF CONTENTS

Exec Summary

Computer science education is a rapidly growing subject in K-12 schools across the nation. Efforts need to bring equity and rigor to a once elective subject in Advanced Placement high school classrooms or career technical education programs as it moves to elementary, middle, and high schools in the United States. As computer science becomes an integral part of standard curriculum, it becomes imperative to expand it as a part of K-12 STEM education.

As computer science takes its place in schools, there needs to be a sustainable pipeline of K-12 educators, teachers and leaders who can provide and support rigorous and inclusive instruction in computer science for all students. A key component of the initial preparation and ongoing development of all teachers is post secondary schools of education. Schools of education have courses which provide critical foundational knowledge to teacher candidates, have faculty who conduct research into the best practices of education for those courses, and help early career teachers pursue additional learning that enables them to grow into master teachers.

Without the support of schools of education, the computer science education teaching force will be difficult to create at scale and continue to be a challenge for K-12 schools and principals. Many current efforts only involve the identification of current in-service teachers who are willing to take on teaching of an additional discipline. For computer science to truly be a core subject in K-12, we must treat the preparation of pre-service computer science teachers the same as other subjects.

In this report, we summarize the discussion of researchers, school of education faculty and leaders, and directors of nationwide organizations and offer concrete recommendations for schools of education to include computer science as a part of teacher preparation, professional development, and doctoral training programs. These recommendations are not meant to be implemented all at once, but the expectation is that schools of education will find synergy with other departments, regional or state efforts to expand computer science education and research.

School leaders, such as principals and superintendents, are looking to hire teachers with requisite backgrounds for computer science instruction in their classrooms. Graduates of programs that prepare teacher candidates with fundamental skills or requisite coursework for computer science certification where it exists will be more competitive in the job market for teaching positions. Faculty and PhD candidates who do research in computer science education will be able to apply for federal grants with a focus on computer science or interdisciplinary STEM education, and potentially bring private sponsorship into the college or university from technology companies and private foundations interested in supporting computer science education.

We offer a set of recommendations as actionable steps that schools of education together with other departments and school districts can take and—with appropriate resources and time available—build the foundation to house computer science education within their faculty, students, and teacher education programs.

Report Recommendations

Section 1: Teacher Education

What Teachers Need to Know About Computer Science

- → Schools of education should include content, teaching methods, and state standards for computer science in any preparation programs for teachers who may have responsibility for those standards.

- → Faculty designing new courses or programs should partner with practitioners to make sure coursework aligns with any relevant offerings in regional schools.

- → Faculty in schools of education should regularly connect with the computer science teacher community of practice events and computer science education research publications in order to include current best practices in coursework.

Teacher Development Models for Computer Science Education

- → Faculty of education can connect with members of the departments of computer science who are running professional development or inservice teacher preparation to identify needs and best practices.

- → Faculty preparing courses for future teachers should review example syllabi or program outlines from other institutions for common practices.

- → Schools of education should make sure teacher candidates have content preparation aligned with national and relevant state standards for computer science education.

Section 2: Computer Science Courses and Content for Educators

Computer Science Education in Teacher Education

- → Schools of education can commit tenure-track faculty lines and resources in collaboration with departments of computer science to support teaching and research in computer science education.

- → Schools of education can hire tenure-track faculty who work in computing education research into STEM education lines and encourage their collaboration with computer science faculty.

- → Schools of education can count computer science education research published in conference proceedings at the same level as journal publications to allow junior faculty to get promoted and tenured.

Standalone Computer Science Education in K-12

- → Schools of education should explore offering

FOCUS AREA 1

What Teachers Need to Know About Computer Science

FOCUS AREA 2

Teacher Development Models for Computer Science Education

FOCUS AREA 3

Computer Science Education in Teacher Education

FOCUS AREA 4

Standalone Computer Science Education in K-12

FOCUS AREA 5

Integrating Computer Science Education into Other Disciplines

FOCUS AREA 6

Computer Science Education in Grades K-8

FOCUS AREA 7

Preparing Educational Leaders to Support Computer Science Education

FOCUS AREA 8

Creating the Computer Science Education PhD Pipeline graduate coursework, certificate programs, or even master's degrees in computer science education. These programs should seek to align with state certification efforts to enable teachers who are new to computer science to obtain any necessary licenses.

- → Secondary teacher education programs need to develop and provide courses for preparing teachers with experiences and background in teaching introductory and Advanced Placement computer science courses.

- → Teacher education programs should hire computer science education specialists, just like they have subject matter specialists for science, mathematics, or literacy topics, who not only teach courses but can also supervise and mentor pre-service elementary and secondary teachers.

Integrating Computer Science Education into other Disciplines

- → Schools of education should identify examples of integrated computer science education to include in coursework for education students in other disciplines. These examples could be sourced from currently funded NSF projects.

- → Schools of education faculty and students should interact with students and faculty from other departments on campus that use computer science to solve problems and explore hypotheses in their own disciplines. Suggestions for this include welcoming speakers to education classes or inviting faculty from other departments to speak in the school of education about their work.

- → Faculty and students from schools of education should seek out case studies of interventions that did not produce learning gains in the multiple disciplines attempted in an interdisciplinary project and discuss what may have inhibited the learning.

Computer Science Education in Grades K-8

- → Schools of education should ensure through programmatic requirements that K-8 teachers receive basic literacy instruction in computer science as well as foundational computer science education pedagogical instruction.

- → Faculty in schools of education should have opportunities to familiarize themselves with relevant K-12 computer science education standards for integration into core pedagogical coursework.

- → Candidates for ELL or special education licenses or certificates should include cases or examples with computer science instruction as a part of their training.

Section 3: Education Leadership and Computer Science Education

Preparing Educational Leaders to Support Computer Science Education

- → Computer science educational leaders should participate in conferences and gatherings of educational leadership communities (including superintendents, principals, etc.). Having computer science educators and researchers in these conferences will bring visibility to practicing educational leaders who are often mentors for new administrators.

- → Faculty in educational leadership programs should use case studies with problems of practice for computer science education implementation. There are numerous examples such as Chicago Public Schools, the New York City public schools, and many smaller city or rural schools with computer science education implementation plans.

Creating the Computer Science Education PhD Pipeline

- → Schools of education and schools of computer science should work together to define opportunities for interdisciplinary coursework for interested computer science education PhD candidates.

- → Current computer science education faculty should propose and seek funding for workshops for leaders of schools of education to promote understanding of quality computer science education publications and CVs for faculty candidates.

Introduction

Computer science education is bursting onto the K-12 education scene in the United States. It has emerged as a bipartisan issue, with both former President Obama calling for a computer science for all initiative, including $4 billion in the proposed 2016 budget for states and cities to offer computing education, and current President Trump directing the Secretary of Education to allocate up to $200 million in grants for computing initiatives in schools. Bolstering these efforts has been an equally strong call to promote computational thinking throughout K-12 education. Computational thinking is defined as "taking an approach to solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing" (Wing, 2006, p. 33).

Currently, there a number of computer science education initiatives in the United States led by non-profit organizations both at the national level (such as, CSforALL, Code.org, Exploring Computer Science, and others) and local levels. For example, New York City has mandated every school to provide computing education to all students by 2025; similarly, the city of Chicago and the state of Virginia are making education in "coding" a requirement. Arkansas has mandated computing education in every high school. The recently passed federal education legislation notably includes computer science in the list of "well-rounded" and critically enriching subjects. States are rapidly developing certification and supplemental/endorsement credentials for computer science teachers.

With the expansion of computing education in mainstream K-12 schools, the current approach of providing professional development for current teachers will quickly fall short of supporting a sustainable pipeline of computer science teachers for the scale many cities and states have committed to. Current methods of preparing teachers often rely on in-service teachers, identifying teachers who currently hold a teaching position in a school, and providing them professional development in a specific program or curricula the school commits to offering. If computer science is going to become a core subject in K-12 education, the education community needs to engage with current methods of teacher preparation not only to produce computer science teachers at scale, but also to update the technological preparation of K-8 teachers to include computer science topics and computational thinking. Furthermore, current and incoming computer science teachers need to be prepared to develop teaching strategies to deal with significant diversity issues (Margolis, Estrella, Goode, Holme, & Nao, 2017)

Teacher education programs at colleges and universities now face several questions regarding the implementation of computing education: Where does computing education fit into existing curricula and organizations? Is computing education more like science education or mathematics education? Does it fit more naturally in educational technology or educational psychology? In which courses should we teach all pre-service teachers about computer science and about teaching computer science? How do we prepare teachers to engage with computing and students from underrepresented groups?

In order to address these critical questions, the authors received a National Science Foundation grant to convene computer science education researchers, leaders from large departments of education, teacher education researchers, and computer scientists in a workshop held in April 2017 to help answer the above questions and more. This report shares the discussion outcomes and recommendations from the workshop group members. Additionally, the workshop talks and panels were recorded and are available at http://computingteacher.org.

A note on terminology used throughout this report: Agreed-upon definitions of computer science or computational thinking for K-12 students are difficult to come by. For the purposes of this document, and with a specific lens on the educational system of the United States of America, we will use the definition of concepts and practices provided by the K-12 Computer Science Education Framework[1], and the Operational Definition of Computational Thinking for K-12 from the International Society for Technology Education (ISTE)[2]. Computational thinking describes approaches to problem-solving and habits of mind that are associated with computer science, most often in application areas beyond the use of computer. Furthermore, computer science, information technology, information systems, and software engineering are all fields under the umbrella term of computing. We have decided to use computer science as the term most commonly understood when we reference students learning about concepts and skills that might fit into any of these fields.

[1] www.k12cs.org

[2] http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf

State of Computer Science In Schools of Education

2013 report released by Computer Science Teachers Association (CSTA) entitled "Bugs in the System" highlighted that CS teacher certification in the United States is problematic (Lang, et al., 2013). Specifically, the report examined how CS teachers get licensed in all 50 states and found that teachers lack adequate pathways to become certified to teach computer science. The report found that only 2 out of 50 states and the District of Columbia (4%) require CS certification/ licensure for teachers to teach any CS course, and only 7 states (14%) require it to teach AP Computer Science. These inconsistent teacher certification requirements have led teachers from other content areas to teach computer science with limited training. The most recent annual survey from the Computer Science Teachers Association found that the majority (57%) of CS teachers also teach additional content areas and the majority of computer science classes (62.3%) are offered through the business, technology, or mathematics department in their schools[3].

The current efforts have focused on in-service teacher preparation through professional development opportunities mainly through a number of CS curricula at the high school level. These professional development efforts are supported by non profits such as Code.org, colleges and universities through National Science Foundation supported projects, or for profit companies who provide curriculum or devices to schools. These efforts on in-service teacher professionals are stop-gap measures as there is a high attrition rate of K-12 teachers, in general, and many of the teachers trained will leave the profession after a few years. Prior research has also found that beginning teachers identify the lack of content knowledge and insufficient support and guidance as key problem areas and challenges in meeting students' learning needs. These factors play a significant role in explaining why 40-50% of teachers leave the profession within first five years of joining the profession (Borman & Dowling, 2008), contributing to the challenge of sustainability for CS education initiatives that rely only on in-service teacher development. Thus, there is a critical need to provide teacher education programs that focus on pre-service teachers to meet the growing demand for teachers who are well qualified to teach computer science.

[3] For access to the annual survey results, go to: https://bit.ly/2DZAfYH

CS education preparation for teachers in the United States today is done almost entirely in-service, and is often situated in higher-education CS departments or provided by industry-sponsored groups (Century et al., 2013). Only a handful of teacher education programs in schools or colleges of education across the nation are focused on preparing pre-service CS teachers, and in those programs few students are enrolled. For instance, in two of the University of Wisconsin campuses pre-service CS teacher development courses are on the roster but neither program has graduated a student in the last five years (Emechebe, 2015). These numbers are by no means sufficient to educate and develop a broad enough pool of teachers for K-12 school districts and ensure long-term viability of CS education in US schools. There is no discipline currently taught across all US schools which is not also taught in pre-service teacher education programs.

2.1 Moving K-12 Computer Science Education into Schools of Education

This report focuses on schools of education rather than departments or colleges of computer science/ computing for setting up CS teacher education. In order to have sustainable computer science in schools, we need schools (or colleges) of education to teach new CS teachers, to research new and better ways to teach CS, and to provide broad and equitable access to CS teaching and learning (Yadav & Korb, 2012). Schools of education typically host teacher education programs charged with providing pre-service educators the necessary coursework, practicum and student teaching, and certification required to prepare to teach an increasingly racially and linguistically diverse student population. This setup provides the following advantages:

- Pre-service coursework typically includes a course of study in learning sciences, multicultural education, and teaching methodology classes for particular subjects and grade bands.
- Pre-service observations and student teaching experience, likewise, arranged in collaboration with local schools, is also intended to be aligned with particular subjects and grade bands of certification.
- Programs work with state agencies to ensure that pre-service teacher candidates successfully pass all state credentialing requirements.
- Non-traditional preparation programs allow teachers to work in the classroom while they complete coursework through school/provider collaborations, and account for approximately 12% of teacher preparation.
- Lastly, in-service programs are often housed in schools of education and provide teachers with the continued necessary professional development.

While teacher education programs have been traditionally responsible for the majority of new teachers across other subject areas and have important knowledge on the scholarship of teaching, few programs have successfully created and sustained robust pathways for future CS teachers (Lang, et al., 2013). The roadblocks preventing more certified CS teachers include: the lack of available CS teaching methodology classes, the lack of access to CS classrooms for student teaching, and the lack of a state teaching certificate in CS. Given their insights on teacher preparation and knowledge of these unique challenges, leaders of teacher education programs are uniquely suited to discuss and explore opportunities to prepare CS teachers.

In addition to pre-service and in-service teacher education and certification, schools of education provide programmatic research on teacher development efforts, curriculum design, classroom implementations, and assessment. They also examine administrative leadership and policy issues connected to introducing new initiatives in classrooms and districts. Educational researchers housed in schools of education have the expertise not only to conduct formative and summative evaluations but also to pay close attention to diversity and equity issues that are becoming increasingly prevalent in the field of computing. This is especially important because students and teachers in under-resourced school districts make up the communities most lacking access to introductory and AP CS courses.

2.2. Developing Strategic and Innovative Relationships with School Districts

We furthermore challenge schools of education to develop build strategic relationships with school districts as they innovate and integrate a new discipline into their teacher education programs, professional development, and research initiatives. Two existing successful efforts in the largest school districts in the U.S.—New York City and Los Angeles—provide examples of how these efforts can be rolled on a larger scale.

The Exploring Computer Science (ECS) curriculum and professional development program began in Los Angeles Unified School District (LAUSD), the nation's second largest school district and has over a decade worth of experience preparing teachers for CS classrooms. The ECS course was created in collaboration with educational researchers, school district curriculum leaders, and high school teacher practitioners, and was successful in reaching tens of thousands of high school students and establishing a professional teacher community. The enrollment demographics of the ECS course in Los Angeles school district reflect the racial and gender demographics of school district enrollment.

Designed intentionally to reach diverse student populations who have traditionally not participated in computing courses, the ECS professional development program encourages teachers to build their inquiry-based pedagogy, equity-based teaching practices, and content knowledge to successfully teach the course to all students (Goode, Margolis, Chapman, 2014). Since many of these ECS teachers do not have an extensive background in either computing or this type of active pedagogy shown in educational research to be effective for teaching students of color and girls, the professional development program takes place over two years. Over 2,500 teachers have completed the professional development program, and the ECS course has been adopted in the top seven major school districts and in several states across the nation.

In addition to the long-standing efforts in LAUSD, the recent announcement of CS4All in New York City has secured CS education as a focus area in the nation's largest school system. The CS4All initiative, which mandates that each of the city's 1700+ schools offers CS to every student at least once within a grade band, is primarily supporting in-service teacher professional development with the funds committed by the public/private partnership. Complementary to this effort is an emerging strategy to provide pre-service or in-service education through the network of public and private colleges and universities in the NYC area.

Working with colleges and universities in the area, CSNYC has helped facilitate the creation of proposals to the NY State Department of Education as programs that would lead to teacher certification. One early partner is Queen's College, the nation's third largest pre-service teacher preparation program with over 7,000 students enrolled in teacher preparation programs. The potential partners include a wide variety of models such as an Elementary School model where CS is a choice for students' area of concentration within currently approved programs, a 5th-year master's program, designed to help students earn a certification in Computer Technology, an extension model where teachers certified in another discipline are able to complete 12-18 credits in both CS content and pedagogy to extend a current teaching certificate, and n "update" model, where current educational technology courses are revisited to remove extraneous content and provide foundations for integrating CS education where appropriate.

These strategic relationships within school districts provide examples of the variety of ways CS education, teacher education programs and professional development can be infused into current schools of education.

Workshop Group Reports From the Computer Science Community

Participants in different working groups discussed issues related to each question and produced the following recommendations:

TEACHER EDUCATION

Focus Area 1

What do Teachers Need to Know About Computer Science?

As CS becomes more integrated in many disciplines and careers, more teachers will need to know about computer science in a way that's relevant to their discipline and teaching practice. Clearly, not every teacher needs to know how to be or how to prepare future software developers. We need to define how CS is relevant in the context of pre-service teacher development.

Focus Area 2

Teacher Development models for Computer Science Education

We need to figure out how teacher development for computer science education fits into existing pre-service teacher education programs.

COMPUTER SCIENCE COURSES AND CONTENT FOR EDUCATORS

Focus Area 3

Computer Science Education in Teacher Education

What are the computer science education concepts that all teachers need to know as a part of general teacher education programs.

Focus Area 4

Standalone Computer Science Education in K-12

Some CS will likely be taught in mathematics or science classes, or maybe English, Journalism, and Economics classes. But some CS will be taught as a standalone class.

Focus Area 5

Integrating Computer Science Education Into Other Disciplines

CS is an important topic on its own, and is also an enabling medium to improve learning in other disciplines. As a kind of literacy, computing will end up being integrated into other disciplines, which creates a challenge for teacher preparation that crosses education programs.

Focus Area 6

Computer Science Education in Grades K-8

Elementary school CS is qualitatively different than secondary school CS. We need to address a wider range of needs and possible outcomes, far beyond Advanced Placement exams.

EDUCATION LEADERSHIP AND COMPUTER SCIENCE EDUCATION

Focus Area 7

Preparing Educational Leaders to Support Computer Science Education

Adding CS into the U.S. K-12 education canon is more than a matter of teachers and curriculum. We also need the political will and the support of leaders in the K-12 education space.

Focus Area 8

Creating the Computer Science Education PhD Pipeline

CS education is a need for K-12, and computing education research is an important subdiscipline which will help us to understand CS learning better and to improve CS teaching. We need a pipeline of PhD researchers to grow this important research field.

TEACHER EDUCATION

Focus Area 1:

What do teachers need know about computer science?

We need more CS teachers if we are going to provide high-quality and accessible CS education for all. An important question for any teacher development program is what these teachers need to know to be effective. The first focus area is defining what teachers need to know about computing.

What is the current state of the field for understanding basic teacher knowledge?

There are competing definitions and priorities for the integration of CS education into K-12 education. Over the last year, the K-12 Computer Science Education Framework (Parker & DeLyser, 2017), the ISTE Standards for Computer Science Teachers (Thomas & Knezek, 2008), and the Computer Science Teachers Association K-12 Standards (Seehorn & Clayborn, 2017) have all been released to offer a national definition of age-appropriate CS for all students in the United States. Many states have followed the national lead and have begun updating or creating state-defined standards for CS education. Although national teacher standards do not yet exist, teacher preparation programs should look to the student-facing standards for clear definitions of CS content that teachers will need to teach to students.

As community norms are being established around student content, there are still few examples of research identifying learning trajectories, or even validating the appropriateness of particular concepts and topics for students of different ages. Research is beginning to develop a better understanding about possible progressions needed at the K-8 level to learn basic CS concepts (Rich et al., 2018) but there is too little empirical work supporting this theory and the theory does not cover all of the content that the standards expect for K-8. Overall, we know little about what students of different ages are successfully learning about CS, where major challenges exist for students, and on how to best support them (see also Guzdial, 2015). If we do not know what and how students are learning, it is hard to estimate what is possible for them to learn, and even harder to prepare teachers to achieve those learning goals.

What does success look like?

A successful model of teacher preparation would involve both the definition of the relevant concepts, skills, and practices for the discipline of CS, as well as pedagogical approaches for teaching these concepts in K-12. Teachers would have access to preparation programs that would prepare them with what they need to be successful, highly-capable CS teachers. We are far from this goal now. Currently, most of the preparation for teachers of CS is in short in-service professional development sessions, with little assessment about the success (e.g., if teachers retain the knowledge, use it, and can apply it in the classroom). More critically, we have not yet defined the CS concepts, skills, or practices teachers need to know.

As in other disciplines, the definition of what content knowledge teachers need to have is defined by the grade band and specific content area focus of each teacher. The appropriate CS minimum content knowledge will look different for an elementary school teacher, a high school Mathematics teacher, or a high school Social Studies teacher. The CS knowledge that a first grade teacher needs to teach is different from what a high school teacher needs to teach Advanced Placement CS Principles.

The working group recommended a clear alignment between the minimum content standards for teachers and the content they might be expected to teach at that grade band. Additionally, if CS education standards are defined for all K-12 students, teacher preparation programs may need to include content from the national K-12 standards while the state and even district school systems catch up to new standards for students.

In addition to CS content knowledge, teachers will also need pedagogical content knowledge (Shulman, 1986). Pedagogical content knowledge is knowledge that a teacher has about how to teach a subject successfully. One part are pedagogical methods, e.g., knowing best practices for helping early elementary school students understand the sequentiality of computer programs. Another part is knowledge of misconceptions. Highly-capable teachers know the challenges that face learners when facing difficult concepts or skills, can identify when students are not successfully meeting those challenges, and know how to help the students to overcome these difficulties (Sadler et al., 2013). Finally, teachers need to know assessment methods to determine if their students are successful learning what they need.

Furthermore, teachers will need an understanding of the diversity issues in CS and learn equity-based teaching strategies. Given the historical, pervasive educational inequities in computer science (Margolis, Estrella, Goode, Holme, & Nao, 2017), it is essential to prepare teachers with the knowledge and skills necessary to broaden access and enact welcoming and inclusive pedagogical practices that support diverse students (Goode, 2007; Margolis, Goode, Chapman, & Ryoo, 2014). This attention to equity issues in computer science education should not be relegated to a separate course, but should also be authentically infused and supported across content and methodology coursework.

What would you say to a teacher who asks, “What do I need to know about computer science?"

A K-12 teacher of CS should be expected to have at least the knowledge of the K-12 Computer Science Framework concepts and practices. The content of an Advanced Placement CS Principles course as well as a college-level CS 1 programming class would be an excellent foundation for the content you might be asked to teach in a K-12 CS course. If you are a subject area teacher (not CS) and are looking to integrate CS within your discipline, you should look to have at least the concepts and practices outlined in the K-12 Computer Science Framework (www.k12cs.org).

Successful teacher preparation programs will inspire comfort and confidence in pre-service teachers. Teachers need to have a sense of self-efficacy, i.e., a sense that they are capable and knowledgeable teachers. The most critical need for a teacher just starting to learn CS is a sense that they can learn what they need and that they can become highly-capable teachers. A teacher who learns a lot but is not confident will not succeed in the classroom. A teacher who is still learning but has confidence in her ability to learn CS can be productive in the classroom.

What is the gap between success and where we are now?

Currently, the lack of a shared definition of minimum content knowledge for teachers is an extreme stumbling block. We have a starting point for understanding the minimum content knowledge for students in the K-12 Computer Science Education Framework (Parker & DeLyser, 2017) and the Computer Science Teachers Association (CSTA) standards (Seehorn & Calyborn, 2017). These two documents are being used to prepare state standards for CS education across the United States, and can represent a foundational content knowledge for teachers.

Teachers need much more than content knowledge. The CS education research community is working on definitions of pedagogical content knowledge (Yadav et al., 2016), good assessment practices, and pedagogical strategies for equity and inclusion. As these concepts become more well defined in the literature, schools of education will need to update course offerings and syllabi in order to stay current with the field.

What are the ***key strategic actions*** that a school of education could take to move this work forward

- Schools of education should include content, teaching methods, and state standardts for CS in any preparation programs for teachers who may have responsibility for those standards.
- Faculty designing new courses or programs should partner with practitioners to make sure coursework aligns with any relevant offerings in regional schools.
- Faculty in schools of education should regularly connect with the CS teacher community of practice events and CS education research publications in order to include current best practices in coursework.

FOCUS AREA 2:

Teacher Development Models for Computer Science Education

Currently, few models exist in the United States for the development of highly-effective computing education teachers, especially focused on CS or computational thinking, within schools of education. As seen in Focus Area 1, What do Teachers Need to Know About Computer Science?, highly-effective computing education teachers know their content area, have high self-efficacy in their ability to teach, and have pedagogical content knowledge about how to teach that content (including pedagogy, assessment, and familiarity with common misconceptions and how to diagnose those). As described in Section 2, The State of Computer Science Education in Schools, much of the development of computing teachers with a focus on CS has focused on inservice professional development efforts. While some of this work is supported by post secondary institutions, it is often housed in schools of CS or engineering. In this section we describe models of teacher development could be adopted by schools of education, and recommendations for quality computing education teacher development programs inside schools of education.

What is the current state of the field for teacher development models?

Although the United States has few models for preparing CS teachers, there are numerous programs for "technology" teachers or educational technology specialists. These teachers may know how to support student learning with educational technology, but may not have the content knowledge or pedagogical content knowledge to teach CS. The distributed model of education spread across 50 states combined with very different implementations of CS education has led to confusion about what is needed to prepare and certify teachers of CS. For example, in some states, CS is taught as an academic subject and can be awarded credit for Mathematics or Science. In other states, CS is a career and technical education (CTE) subject and therefore has a completely different set of certification requirements, sometimes including technical work experience. This lack of shared understanding of the course pathways in CS school programs has led to confusion about how to prepare teachers to teach computing.

What does success look like?

Successful models of teacher development in any subject include a focus on pedagogy, content, and pedagogical content knowledge specific to the domain (Shulman, 1986). Schools of education already offer pedagogical courses for pre-service teachers and graduate students of education in other disciplines (e.g., reading, mathematics, and science). In addition to methods courses, quality programs in computing education would include content specific and pedagogical content specific coursework.

In content area courses for computing teacher development this means including content area courses in CS as a part of the required core. These courses do not have to be courses designed for CS majors, but do need to cover the content area standards teachers will be expected to cover in their own classrooms. Schools of education can look to the K-12 CS framework[4] or individual state standards (where available) to determine the appropriate CS concepts for each grade band.

For teachers preparing for elementary school, additional CS coursework may not fit into the required core. However, many schools of education require a technology course as a part of teacher development. In the section “CS Education in Teacher Education” this report describes examples of integrating CS and computational thinking into the general preparation of all teachers.

The pedagogical core for CS education needs to broaden to include issues related to inquiry and project based teaching and learning, examples of high quality CS teaching, the need to broaden participation and pedagogies that support (and hinder) diverse participation in advanced pathways, and finally, career profiles so teachers can present the diverse ways in which computing professionals impact the world.

What is the gap between success and where we are now?

The gap is wide, but hard to measure. Some states understand the needs of computing education, while other states have not yet started considering how to teach computing in their schools. What is clear that no US state currently has a system where schools of education use best practices to prepare CS teachers with content and pedagogical knowledge. Closing that gap will require both landscape and definitional work.

First, the community needs a comprehensive survey of CS teacher preparation, including what is

[4] K12CS.org

Identifying the Needs of Teachers

At the writing of this report, both the Computer Science Teachers Association (csteachers.org) and the International Society for Technology in Education (ISTE) are working on tools to support teachers in the identification of appropriate content and skills for computer science instruction. Together with CSforALL, who welcomes schools of education with teacher preparation programs as members, these organizations will be important resources for schools of education looking to develop content for courses.

taught, where it is taught, and to whom. As much of this work is currently happening in professional development sessions, the in-service teacher learning must be included in such a survey. Additionally, such a survey should ask the educators themselves how they combine preservice, inservice, and professional development opportunities to grow their professional practice. Additionally, such a landscape report needs to acknowledge the diversity of preparation pathways, since many CS teachers currently come to the discipline as a secondary (or even tertiary) subject or career. These diverse pathways may not be the best to emulate, but they can provide insight into the types of experiences that are sought after as novice teachers seek to gain mastery.

Secondly, there needs to be clear alignment between student standards and teacher content preparation. As the definition of what is appropriate content for K-12 students is becoming clear though state and national frameworks and standards, teacher preparation programs need to be aligned. Early CS courses in college often focus on the programming aspect of CS, yet K-12 standards and frameworks also include networks and the internet, systems, data, and the impacts of computing. Teachers will need a broad introduction to all of these concepts if they are to be expected to teach them to students.

In other disciplines, the content and methods courses can draw upon a rich collection and history of examples, case studies, research, and practice documents. As a new subject, CS education needs the development of these items for teacher preparation. Many of the professional development providers have created resources that would be useful to schools of education, and the recommended landscape should include links to appropriate resources that will aid the creation of content and methods courses.

What are the *key strategic actions* that a school of education could take to move this work forward

- Faculty of education can connect with members of the department of CS who are running professional development or inservice teacher preparation to identify needs and best practices
- Faculty preparing courses for future teachers should review example syllabi or program outlines from other institutions for common practices.
- Schools of education should make sure teacher candidates have content preparation aligned with national and relevant state standards for CS education.

COMPUTER SCIENCE COURSES AND CONTENT FOR EDUCATORS

Focus Area 3:

Computer Science Education in Teacher Education

We want education faculty who focus on CS education in their research and in support of teacher development to be successful. They need to be recognized for their contributions and be tenured and promoted for those contributions. This focus area looks at how to promote the field of CS education within schools of education.

What is the current state of the field for CS education within teacher education?

Schools of education face a number of challenges in terms of preparing more CS teachers. Trends over the last decade have shown a general lack interest in pursuing a teaching career. In a 2016 national survey, The National Education Association reported that the number of students planning to major in education in 2014 dropped to an historic low of 4.2%. This is compared 11% in 2000 and, 10% in 1990, and 11% in 1971 (Flannery, 2016). Given the declining enrollment in schools of education, developing and maintaining new programs to train CS teachers faces an uphill task.

Growing CS education in teacher education also faces another hurdle. There is a lack of understanding of what CS entails and sometimes confound computer literacy for CS (Wang, Hong, Ravitz, & Hejazi Moghadam, 2016). As such teachers and school administrators are unaware of and unprepared for the requisites of high quality CS. The weak market for preparing teachers to teach CS has been exacerbated by the constraints of the system to train teachers, which limits the amount of time, costs, and credits available for aspiring teachers to explore additional teaching certifications, such as CS.

Computational Thinking Course at Michigan State University College of Education

At Michigan State University College of Education, preservice teachers are introduced to computational ideas and tools as a part of their Introduction to Technology course. The course shifted from a focus on educational technology tools in the classroom exposing future teachers to core computational thinking concepts and practices. The next step in this work is to integrate computational concepts within content area methods courses to allow preservice teachers to understand how they apply within particular K-12 subject areas.

While these two looming issues are pervasive across the education landscape, the consistent lack of supply and demand for trained teachers continue to define the scarcity of CS teachers. With not enough positions in K-12 schools to hire full-time CS teachers, there is little incentive for schools of education to develop standalone programs to certify CS teachers. Typically CS teachers also teach other subject areas, which are considered to be more recognized, such as: math, science, business management, and technical education. Thus, CS licensure is an add-on to a primary teaching license, such a mathematics.

Given these constraints, what kind of knowledge do CS teachers need to teach computing ideas to students at the elementary and secondary level? We believe there must be a relentless focus on both the knowledge of the content itself as well as an understanding of how to make CS ideas comprehensible to students what Shulman (1986) called pedagogical content knowledge. It is also important to recognize that the knowledge one needs to learn to become a programmer is different from that which you need to teach CS (Guzdial, 2014).

What does success look like?

There are several models which could address the challenge of producing a pipeline of qualified CS teachers. One model is a standalone four-year program. In addition to the challenge of sustaining a

standalone program with low enrollment, there is also the possibility of a student graduating with a Computer Science minor pursuing a career outside the teaching profession with competition from employers willing to hire and pay more than the average annual salary for teachers, which is $57,000 according to the Bureau of Labor Statistics. While a standalone model provides the best way to train CS teachers, we acknowledge the challenges in developing such programs as discussion previously. So what ends up happening at most schools is a supplemental licensure program where CS teachers have a primary area of certification and then pursue additional courses to get certified to teach CS. This is especially true for training secondary teachers (grades 6-12), where CS can be offered as a standalone subject. Regardless of the pathways to train teachers to teach CS, we believe that there need to be a core set of courses preservice teachers need. We believe that teachers need at a minimum courses on CS content (such as, introductory programming and data structures) and pedagogical content knowledge such as the CS methods course described by Yadav and Korb (2012).

What is the gap between success and where we are now?

One of the challenges particularly relate to grades K-5 (elementary) within K-12 systems. Elementary teachers focus solely on the primary subjects which are tested without having a strong knowledge base of any one particular subject. In primary schools, math and literacy are the two areas that are heavily tested at the state level to measure the proficiency of K-5 students. As a result, CS as a standalone subject might not be a realistic expectation for all primary schools; however, incorporating the basic concepts of computational thinking into classrooms and preparing teachers to teach computational content are realistic. Computational content around problem solving, creative thinking, algorithms, computer programming, and general principles from CS are examples of programmatic aspects which could be taught by primary school teachers. The main hurdle in primary schools is around preparing teachers to incorporate these ideas into their classroom activities. How are we going to get teachers to focus on CS when they are already so pressured on teaching their kids to be proficient in so many other subjects?

The best way for computer science is to be integrated it into other subject areas, such as mathematics, science, and literacy. But the challenge there becomes that you need to educate preservice teachers to embed computer science and computational thinking within those content areas. This requires elementary teacher education faculty to be trained to themselves understand computational thinking ideas and how they could be integrated within teacher education courses. We believe that preservice teachers can be exposed to computational thinking ideas as a part of intro educational technology courses, which majority of preservice teachers take during their program. Then, they could learn to apply those ideas within the context of a subject areas as a part of their methods courses (Yadav et al., 2016). However, this would require ongoing collaborations between elementary teacher education and CS educators to compensate for the faculty turnover in teaching these courses.

What are the ***key strategic actions*** that a school of education could take to move this work forward

- Schools of education can commit tenure-track faculty lines and resources in collaboration with departments of CS to support teaching and research in CS education.
- Schools of education can hire tenure-track faculty who work in computing education research into STEM education lines and encourage their collaboration with CS faculty.
- Schools of Education can count CS education research published in conference proceedings at the same level as journal publications to allow junior faculty to get promoted and tenured.

Focus Area 4:

Standalone Computer Science Education in K-12

As a developing academic discipline, the models of CS in K-12 is still formative. Some schools and districts are choosing to offer CS as a standalone subject—either as a course in middle or high school or as a pull out technology subject in elementary—while others are looking for ways to integrate CS with other existing curriculum areas (see also next section). In this section, when we refer to standalone CS education, we are referring to instructional time where instruction focuses primarily on CS content. There may be additional content areas addressed by examples in the instruction, but there would not be explicit instructional goals for those additional content areas.

What is the current state of standalone CS courses in K-12?

A majority of standalone CS education have concentrated on AP CS or introductory overview courses on the high school level where subject-matter specific courses are more common. Only few stand-alone courses and programs exist for middle and elementary schools (Hubwieser, Armoni, Giannakos & Mittermeir, 2014). This situation in CS education in the US is reflective of efforts in many other countries. For instance, countries such as the Germany, Israel, New Zealand, Scotland and the United Kingdom also have standalone CS courses for the high school students, while other countries such as South Korea have decided to develop stand-alone courses only for the elementary and middle school grades (Hubwieser et al., 2015).

Elementary schools tend not to divide the day into courses, but instead include the opportunity to focus on specific subjects, or have pull out sessions for different disciplines, like music or art. CS education is being implemented as a standalone subject in many elementary schools across the country. Teachers are either being asked to devote a particular amount of instructional time to CS specific content and practice, or technology or library media specialist teachers are using some of the pullout time to complete CS lessons.

Teacher preparation in this area is primarily provided by the companies, organizations, or researchers who have produced the curricula or materials the teachers will use in their courses (see also section 3.2). There are few programs in schools of education that prepare generalist CS teachers, that is a CS teacher who could step into a stand alone CS course and pick up and teach the material. This is very different from the preparation of any other disciplinary teacher in K-12. In most cases, teachers have college-level coursework in the academic subjects they will be asked to teach, and use that general knowledge as a foundation for adopting and adapting curricular materials. Any professional development focuses on the specific learning objectives and pedagogies for new approaches and not on the basic K-12 level concepts themselves. In CS, however, much of the professional development is spent with teachers completing the activities expected of students to build the basic content knowledge of CS.

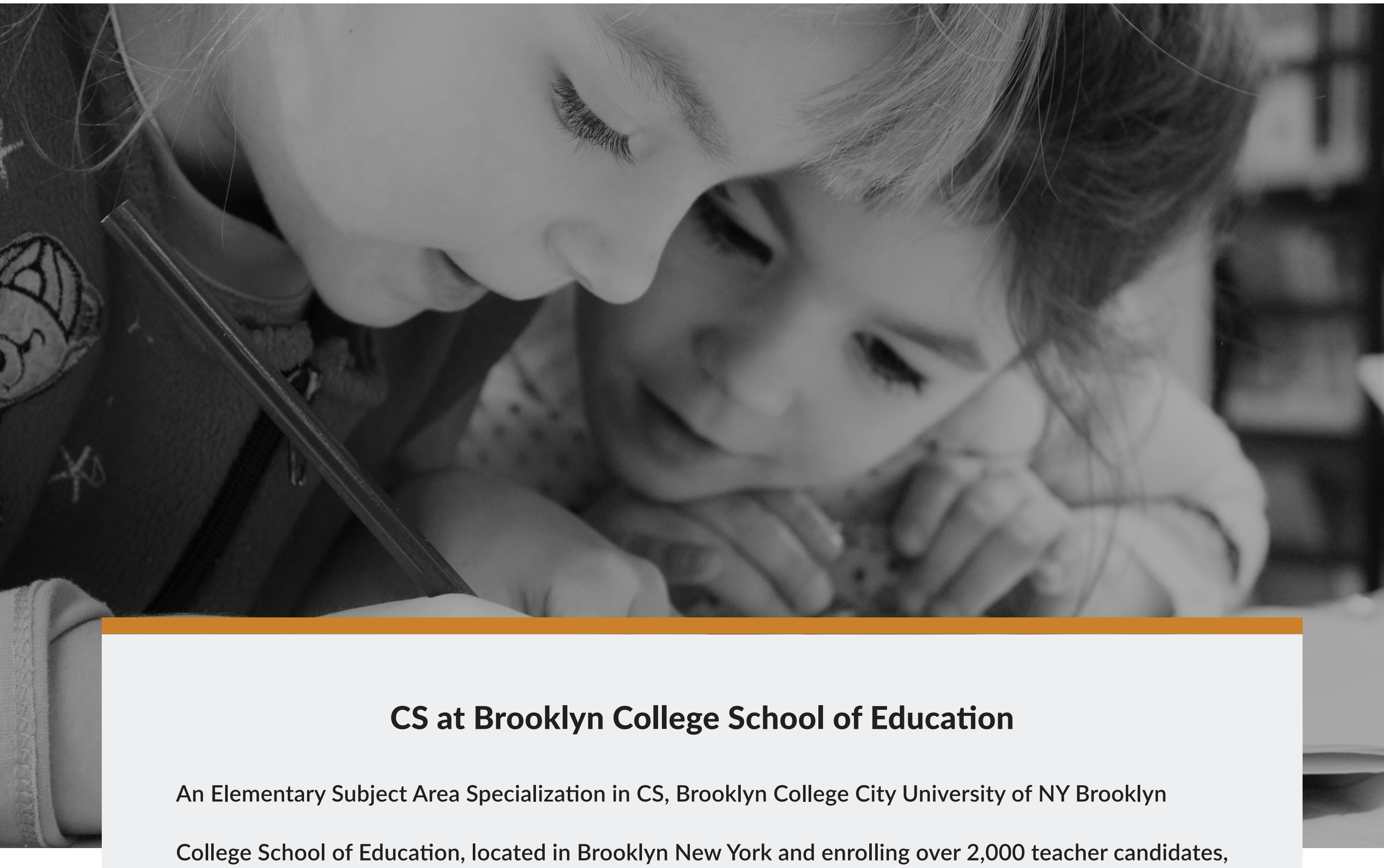
What does success look like?

On the high school level, several standalone courses are available and have been offered for many years nationwide. The most prominent and popular example of a standalone course in CS education is the AP Computer Science Principles A, a college-level semester-long course taught in many high schools in preparation for the AP exam.

The following courses have also been endorsed by the College Board as AP equivalent: (1) Beauty and Joy of Computing which uses the SNAP programming language, (2) Mobile CS Principles which uses the App Inventor for the Android programming language, (3) UTeach CS Principles which uses Scratch processing, (4) CS Principles which uses App Lab and Javascript, and (5) Project Lead the Way which uses App Inventor and Java programming languages. The curriculum for these courses is available to download for free under the Creative Commons license (except for Project Lead the Way). Many of the programs or organizations that have developed these courses also offer professional development for high school teachers wanting to use these courses in their school.

In addition to AP courses, introductory courses are also offered on the high school level. For instance, the Exploring Computer Science program provides a year-long overview on various CS topics such as human-computer interaction, problem solving, Scratch programming, data science, web design and, robotics for the high school level. The top seven school districts in the country as well as several states have adopted this course for their high schools. Likewise, the Programming, Games, and Apps Society has been used extensively in the State of Georgia and provides various modules that introduce students to app design and related issues.

On the elementary and middle school level, there are fewer widespread examples of standalone courses. Examples of standalone CS content that may be taught by a teacher include Scalable Game Design provides a sequence of game design tasks that introduce students to key computational concepts and practices. Furthermore, Code.org has developed



CS at Brooklyn College School of Education

An Elementary Subject Area Specialization in CS, Brooklyn College City University of NY Brooklyn College School of Education, located in Brooklyn New York and enrolling over 2,000 teacher candidates, has an elementary education subject area specialization for CS. This is a legacy program that has existed for 15 years, and the college is now working to revise the required coursework and update the program to serve teachers who will be prepared with college level CS coursework and appropriate pedagogical courses.

CS programs for the middle school and elementary school levels that provide a broad introduction in various CS topics. Either of these programs can be taught in a semester-long or a year-long course format. Furthermore, the ScratchEd Team at HGSE has developed the Creative Computing guide that outlines a series of introductory programming projects. The MakeCodePlay guide describes the design of games and controller activities on and off the screen to introduce students to various computational thinking concepts and practices.

For schools and colleges of education who wish to prepare CS specialist teachers, collaboration with the department of CS at the college or university would be an excellent first step in establishing such a program. There may be legacy course titles or numbers that are pedagogical courses in CS education, for example the City University of New York identified a Teaching of Programming in Basic course that was in the course listing however had not been offered for 15 years. Lehman College, a CUNY school, reinvigorated the course and now offers Teaching of Programming in Scratch for CS credit for students in the school of education[5].

What is the gap between success and where we are now?

Schools and districts should have the ability to choose between stand alone CS education and CS offerings that are integrated into other disciplines, especially at the elementary level. With the ability to choose, the gaps come not in trying to get to 100% of schools offering stand alone CS, but instead in looking for equity in existing courses and availability of teacher preparation and ongoing development to staff those courses.

With regards to equity, the most significant gaps can be seen in the relative populations of students enrolled in stand alone electives when compared to either the overall United States population or even the population at individual schools (Margolis et al., 2017). In the past, mostly well resourced schools, such as private schools, magnet city schools, or suburban schools, have been able to offer standalone CS courses which are often Advanced Placement courses, thus resulting in significantly higher percentages of White and Asian students enrolled than the national population[6]. In contrast, high school students attending neighborhood or rural schools often have had no access to Advanced Placement CS courses (Ericson & Guzdial, 2014). Although Advanced Placement Computer Science is not the only standalone CS course, it is the only one for which national statistics are available. Girls are also significantly underrepresented in the Advanced Placement CS courses, often representing less than 20% of exam takers in the Advanced Placement Computer Science A course.

Recently, programs like Exploring Computer Science which over the last ten years have reached in LAUSD alone over 14,000 high school students of color and the new Advanced Placement CS Principles courses have significantly increased the diversity of schools where CS is taught as a standalone subject, and the diversity of the student body taking those courses.

[5] Further information and links to the above programs and courses can be found in Appendix A

[6] This data is available at: https://tabsoft.co/2IWeSv3

In order to continue to address the issue of equity, schools will need the ability to hire CS teachers from teacher candidate pools, instead of having to spend precious resources to provide in-service development to existing teachers. There are few teacher preparation programs available that allow for the rigorous preparation of CS teachers at a level where an Advanced Placement course is not outside of their content knowledge. Similar to a high school math teacher who could teach Advanced Placement Calculus, the equivalent of college mathematics 101, we should expect a high school CS teacher to have at least the equivalent knowledge of a college-level Computer Science 101 course.

In addition to preservice options, there is a large number of teachers who attended in-service professional development to begin teaching a specific course or curriculum for CS. At the moment there are little to no options for those teachers to continue to seek content and pedagogical knowledge either through master's degrees in CS education or through individual advanced coursework. Disciplinary teachers, such as Math or Science teachers, have the ability to return to schools of education for master's degrees or advanced certificates to increase their content and pedagogical cores, yet there are very few options for CS teachers to do the same. As many states are considering CS teacher certification, these programs will soon become necessary for CS teachers.

What are the ***key strategic actions*** that a school of education could take to move this work forward

- Schools of education should explore offering graduate coursework, certificate programs, or even master's degrees in CS education. These programs should seek to align with state certification efforts to enable teachers who are new to CS to obtain any necessary licenses.

- Secondary teacher education programs need to develop and provide courses for preparing teachers with experiences and background in teaching introductory and Advanced Placement CS courses.

- Teacher education programs should hire CS education specialists, just like they have subject matter specialists for science, mathematics, or literacy topics, who not only teach courses but can also supervise and mentor pre-service elementary and secondary teachers.

Focus Area 5:

Integrating Computer Science Education into other Disciplines

CS can be taught in schools as a separate subject, or it can be integrated into existing disciplines. There are tradeoffs in either choice. Integration might be less effective, but might reduce how much CS students learn. This focus area considers explicitly how we might integrate CS education into existing school classes.

What is the current state of integrating CS education into other disciplines?

As computer science increasingly impacts more of the world, there are numerous opportunities to integrate CS with other academic disciplines in K-12.

In addition to the real world growth of interdisciplinary CS integration, schools are also being asked to take more problem-based approach to education, often involving several disciplines in activities or assignments for students (English, 2016). We see

many elementary schools integrating CS into other disciplines, sometimes as part of statewide mandates. We see many high schools teaching CS as a separate subject, but may still be integrating CS into some subjects, such as Algebra 1 (Schanzer, Fisler, & Krishnamurthi, 2018). We are not taking a position on which is preferable. Rather, we see both integration and separation being common across the states. This section is exploring the integration issue explicitly (refer to the previous section for CS as standalone course).

Science, engineering, and mathematics disciplines are often the first to come to mind when considering the integration of CS education into K-12 subjects. Historically, K-12 CS education started with mathematics and CS being combined for early geometry learning by Seymour Papert in the 1980's (Papert, 1980). The Next Generation Science Standards (2013) also include a cross-cutting concept of computational thinking that is well suited to integrated CS education and science education units. In addition to integrated units with mathematics and science, some programs also include alignment with standards for history, art, or english language arts as well.

Although the idea of multidisciplinary education with transfer to multiple domains sounds promising, CS instruction is not a magic bullet that will somehow promote learning in multiple disciplines.

CS Integration in Schools-Programs that Work

One of our best examples of CS integrated into other subjects in today's schools is Boostrap: Algebra where students learn Algebra through building a video game. The curriculum is designed to scaffold students through bridging the worlds of programming and algebra. The assessment results show student improvements in solving mathematics word problems, even without scaffolding (Schanzer, Fisler, Krishnamurthi, 2017).

Successfully integrated CS activities have learning goals and either explicit instruction or explicit reflection for both domains to promote learning. A student may create a simulation depicting viral growth in a population, however without instruction to understand the science behind the simulation or a reflection that enables the student to connect the behavior on the screen to the desired science instructional goals, very little domain learning will occur.

Research into integrated CS education curricula often highlights not only the appropriate alignment between the CS content and the integrated domain, but also the ways in which that alignment promotes learning in both disciplines. There is room for additional research by faculty in schools of education to explore these interdisciplinary connections.

What does success look like?

Successful interdisciplinary CS education requires that teachers have conceptual understanding from both CS and the domain they are integrating with. Although some integrated curricula and activities are available, teachers ultimately will be responsible for conducting the lessons, assisting students with the activities and assessing student knowledge, ideally in all domains. As teacher preparation programs seek to prepare teachers for the types of instructional settings they will encounter once employed, examples of interdisciplinary work involving CS could be integrated into discipline specific teacher programs.

Case studies involving CS integration with mathematics and science would be included as a part of methods and pedagogy courses for these disciplines. Teachers would be prepared with at least the basic CS conceptual knowledge expected of high school seniors, and schools of education could work with schools of CS to help create introductory CS courses to bridge the gap between the current generation, which may have little to no exposure to CS in K-12, to a time when all students enter college with basic foundational knowledge in CS.

In addition to coursework, opportunities for students enrolled in schools of education could include chances to see the type of integrated CS happening at the college or university CS department. Inviting guest speakers or perhaps a panel of faculty who use CS to work on multidisciplinary projects would provide excellent opportunities for education students to see the real world examples of integrated CS.

What is the gap between success and where we are now?

As an emerging discipline in K-12, CS education does not share the common knowledge of other academic subjects. Every teacher, regardless of subject area of specialization, has taken math or history as a part of their academic experience and so has a general contextual understanding for what topics are taught in K-12 and how they might integrate with each other. In contrast, CS is unique in that there is no shared understanding of even the most basic principles (imagine if the general population had never been taught addition). To close this gap, we must provide foundational understanding for all teachers in CS and computational thinking, as well as give examples and case studies of what integrated CS will look like.

Schools of education will not only produce students with the requisite knowledge of interdisciplinary projects that include CS, but existing faculty will also broaden their experience to include interdisciplinary activities with CS as well. When "Writing Across the Curriculum" (McLeod & Soven, 1992) was a national movement, faculty within math education and science education departments were able to offer suggestions to future teachers about how to integrate writing into their courses. Without this shared understanding, faculty will not be able to offer similar guidance to future teachers regarding integrating CS education into other subjects. All faculty in schools of education should be provided opportunities to engage with relevant CS education standards and observe professional development or classrooms where CS is being integrated into other disciplines.

What are the ***key strategic actions*** that a school of education could take to move this work forward

- Schools of education should identify examples of integrated CS education to include in coursework for education students in other disciplines. These examples could be sourced from currently funded NSF projects.
- Schools of education faculty and students should interact with students and faculty from other departments on campus that use CS to solve problems and explore hypothesis in their own disciplines. Suggestions for this include welcoming speakers to education classes or inviting faculty from other departments to speak in the school of education about their work.
- Faculty and Students from schools of education should seek out case studies of interventions that did not produce learning gains in the multiple disciplines attempted in a interdisciplinary project and discuss what may have inhibited the learning.

Focus Area 6:

Computer Science Education in Grades K-8

Primary and middle schools are particularly interesting areas for including CS education. We have less experience and research support for K-8 CS education, but the potential impact and benefits for CS education in K-8 are significant. This focus area considers the specific challenges to promoting CS education in K-8.

What is the current state of the field of CS in grades K-8?

CS education is experiencing a resurgence in the K-8 grade band. With the inclusion of computational thinking standards in the Next Generation Science Standards (2013), the Hour of Code campaign with K-8 resources[7], and the inclusion of concepts and practices in the K12 CS Education Framework[8], elementary and middle school teachers have more information about teaching CS in their grade bands than ever before. CS education is resurfacing in elementary schools, after the programming languages Logo and Basic were taught and then disappeared from elementary classrooms in the 1980s and early 1990s.

The extension of modern CS from college and high school, need to take into account special factors in grades K-8. First, students in the early grades do not yet have the abstract reasoning capabilities to learn CS the same way as high school or college students. Offering CS in the early grades is about identifying the core fundamentals of computing and providing students with concrete experiences and reflections that solidify those foundations before they explore more advanced concepts (Kafai & Burke, 2014).

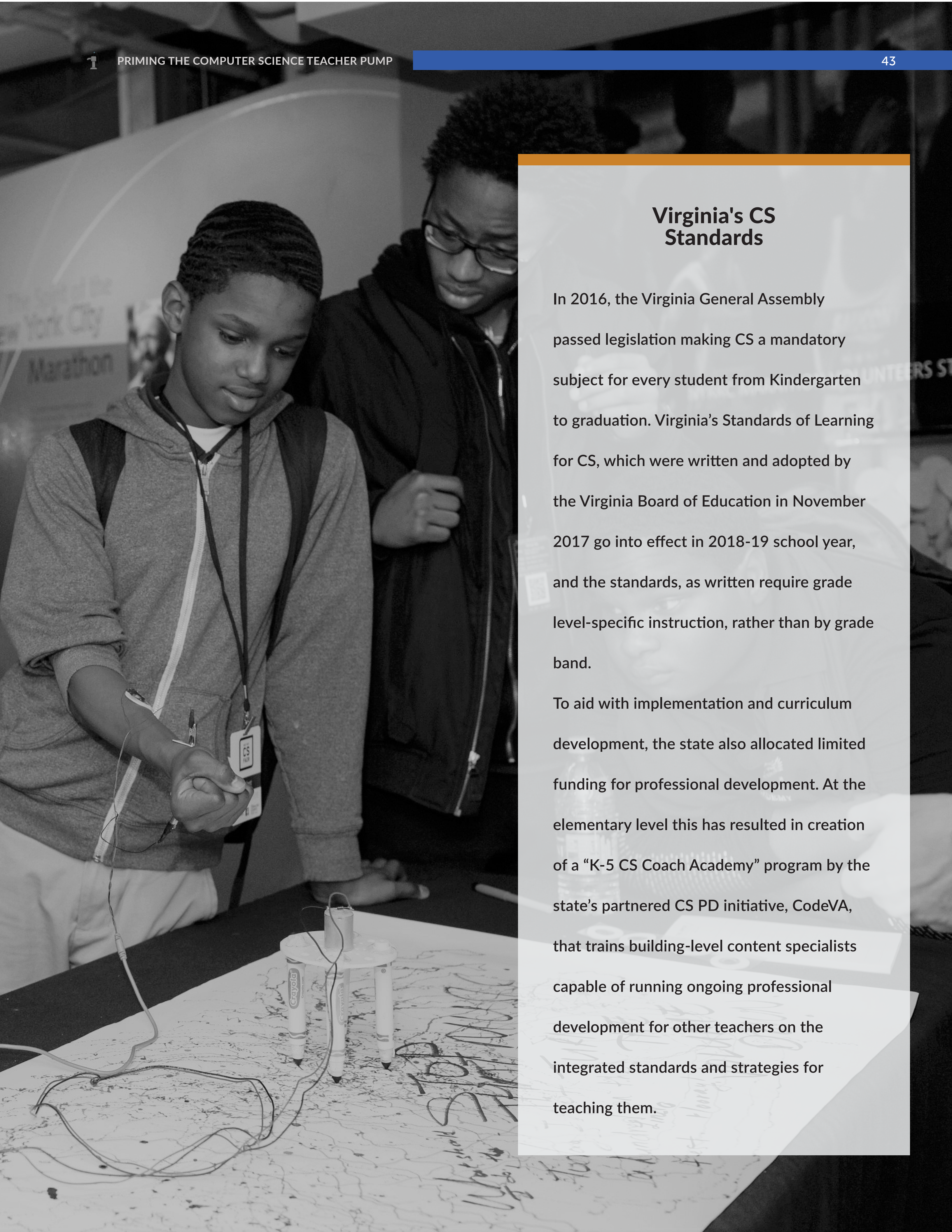
Currently, there are a large number of curricula available for students in grades K-8[9]. These curricula vary widely in the content they cover, the pedagogical activities they use to deliver that content (online puzzles, robots, offline activities including worksheets and board games), and the integration with other disciplines (math, science, ELA, etc.). Currently many state departments of education are also rewriting state standards to include CS.

In addition to high level definitions of what content and curriculum is appropriate for students in

[7] www.hourofcode.com

[8] www.k12cs.org

[9] The CSforALL Consortium lists over 115 curriculum providers with K-8 content on www.csforall.org/members at the writing of this report.



Virginia's CS Standards

In 2016, the Virginia General Assembly passed legislation making CS a mandatory subject for every student from Kindergarten to graduation. Virginia's Standards of Learning for CS, which were written and adopted by the Virginia Board of Education in November 2017 go into effect in 2018-19 school year, and the standards, as written require grade level-specific instruction, rather than by grade band.

To aid with implementation and curriculum development, the state also allocated limited funding for professional development. At the elementary level this has resulted in creation of a "K-5 CS Coach Academy" program by the state's partnered CS PD initiative, CodeVA, that trains building-level content specialists capable of running ongoing professional development for other teachers on the integrated standards and strategies for teaching them.

K-8, we also need clear examples and case studies of quality CS education. Schools of education use such case studies for foundational methods courses and to discuss pedagogical approaches in preservice teacher education. These examples and case studies can include media or video content, and should be aligned to general best practices for appropriate age groups for students.

What does success look like?

Successful implementation of K-8 CS education would require all K-8 teachers to have a fundamental understanding of key CS concepts and practices relevant for this age range. The premise of CS for all is that these fundamental concepts are accessible for all students, and therefore preservice teachers should have the ability to learn content we are expecting of all 8th grade students. Schools of education need to infuse examples into general methods and pedagogy courses to help teachers become familiar with not just the CS content, but also what CS looks like in an elementary classroom.

In addition to covering CS concepts, quality CS education includes computational thinking practices (recognizing problems, creating artifacts, testing and refining solutions) as well as explicit practices around collaboration, communication, and inclusion. From early grades, teachers need to support and scaffold the practices of CS education and help students develop productive and inclusive habits that translate into later more complex team projects.

Finally, teachers need to be prepared with strategies to help struggling and at risk students. Teacher prepared for special education services would also have strategies and scaffolds for students with disabilities. No students would be left without support, including those for whom English is not their first language. These preparations would be part of the generalist training of a teacher, just as a Mathematics or English teacher is prepared to support a wide range of learners in their classes.

What is the gap between success and where we are now?

Currently, teachers receive very little preparation to teach CS in grades K-8. Instead, professional development opportunities seek to close the gap in the current teaching workforce and ensure that we do not leave an entire generation of students without any CS instruction. Schools of education could look for examples from CS education researchers and awardees from the National Science Foundation STEM+C program[10] to identify relevant activities and cases for use in traditional subject area coursework.

[10] Recent awards made through the program available at https://bit.ly/2uhQ7WO

Curriculum providers are also working to integrate CS into disciplinary resources, however teachers will need to assess those materials to see if they are relevant and meet requisite learning objectives for students. Schools of education can help teachers prepare for this by presenting examples of not only good implementation, but superficial integration as well in order to prepare teachers to be critical consumers of content produced for their classrooms.

To be in line with many K-8 school leadership structures, opportunities should exist for teachers to extend beyond introductory professional development to explore mastery building coursework. In addition to equipping classroom teachers with basic CS knowledge, schools may eventually need CS coaches, directors of curriculum and instruction for CS education, and other leadership candidates for such positions.

Finally, teachers will need to modify or scaffold materials for diverse learners. Teacher candidates who are pursuing certificates or extensions for ELL or special education students should also have the ability to support CS teachers in the scaffolding of materials for diverse learners.

What are the *key strategic actions* that a school of education could take to move this work forward

- Schools of education should ensure through programmatic requirements that K-8 teachers receive basic literacy instruction in CS as well as foundational CS education pedagogical instruction.
- Faculty in schools of education should have opportunities to familiarize themselves with relevant K-12 CS education standards for integration into core pedagogical coursework.
- Candidates for ELL or special education licenses or certificates should include cases or examples with CS instruction as a part of their training.

EDUCATION LEADERSHIP AND COMPUTER SCIENCE EDUCATION

Focus Area 7:

Preparing Educational Leaders to Support Computer Science Education

As we think about educational leaders, we recognize there are several permutations of a CS leader in education. For example, CS leaders can be classroom teachers, district superintendents, school system leaders, and principals. Since the important role of teachers and teacher education are discussed elsewhere, this section will focus on those considered to be serving in formal educational leadership positions school, district, and state-level administration.

What is the current state of the field for preparing educational leaders to support CS education?

There is urgency around preparing administrators and other educational leaders with the knowledge and skills needed to support CS teaching and learning for all students. Recent data indicates that CS is not a priority to administrators, despite strong support from school communities, students, and teachers (Wang et al., 2016). Additionally, many administrators may not understand the difference between traditional technology education which include word processing, spreadsheets, presentation software and keyboarding, and CS education which include the design and analysis of software, algorithms, or digital artifacts, often with code.

Administrators and other educational leaders must not only have an understanding of what content constitutes CS education (and what does not), they must also be aware of the trade-offs between stand alone CS education, integrated CS education, and how to create rigorous pathways for students across multiple grades, both for general student CS literacy and for college preparation for students who would like to explore STEM disciplines with more rigor.

To successfully do this, CS education must be fully established within the complex and multi-layered United States school system. Building a robust and cohesive CS Education ecosystem for students requires the support of all stakeholders within this complex system of educational leaders, such as policy-makers, superintendents, and principals. Many schools of education have educational leadership programs for mid-career professionals that include such information in their courses. Currently, there are few—if any—options for administrators to learn about CS education pathways.

Additionally, administrators often are part of the teacher feedback and evaluation cycle and need an understanding of what quality CS pedagogy looks like for teacher feedback purposes. The most common option now are the state-level convenings offered by many state working groups for CS education

(Guzdial, 2016). In these workshops speakers share best practices, offer resources, and provide important information for both educational leaders and teachers.

What does success look like?

Among educational leaders, superintendents are especially critical, because they can influence district-level strategic plans in a powerful way. For example, superintendents have the autonomy to draft and push forward strategic plans that will ultimately advance technology throughout their districts. Winning the support of superintendents would be imperative as they can leverage their decision-making power to impact practices adopted throughout the rest of their district. According to a recent Google poll (Wang et al., 2016), only 30% of superintendents think CS education is important, compared to 50% of teachers

Engaging School Leaders in CS Education with CSforALL

CSforALL[11] is using a process and toolkit called the SCRIPT (Strategic CSforALL Resource and Implementation Planning Tool) to help schools create implementation plans for CS education. The tools include rubric areas for leadership, as well as providing support for administrators who supervise or evaluate teacher performance. To successfully prepare administrators for these roles, schools of education can support administrator knowledge through the use of case studies of CS education implementation and supervision.

and building principals. This statistic highlights the opportunity to improve the overall sentiment and acceptance of superintendents to drive forward district-level adoption of CS preparation. While there is already greater approval among teachers, invoking buy-in of superintendents will create a strong combination of a bottom-up and top-down approach to driving forward CS preparation across districts. Reaching in-service educational administrators will require that professional convenings of regional educators include programming on CS education.

What is the gap between success and where we are now?

In schools of education, educational administration licensure and degree programs are often curricularly and programmatically separated from the teacher education and curriculum and instruction programs. Many administrative programs offer a common set of classes required to become an administrator: school law, finance, leadership, and others that are particularly targeted towards the skill-set required of administrators. There is often little room in the curriculum for administrators to consider particular content-specific support of teaching and learning for any subject, including CS. Yet, many of these programs offer opportunities for educators to examine "problems of practice" from an administrative perspective as part of a doctoral thesis, and there are already emerging examples of how CS learning opportunities can be infused as possible areas of study for administrative licensure and degree programs.

In addition, it is notable that within schools of education, Deans are important stakeholders for

[11] www.csforall.org

preparing new administrators through educational leadership programs, as they set the fundraising priorities and importantly, are a key part of the powerful alumni networks associated with administrator licensure programs.

There is a need for a framework that maintains the authenticity of the CS community. While garnering support from key stakeholders within the schools of education and educational leadership communities are important goals, there also is a need to ensure all educational stakeholder groups are strategically aligned with the main priorities of preparing and teaching CS. Specifically, strategic priorities would align stakeholders to design a cohesive framework for:

- Messaging
- Research, supporting evidence-based effective practices, and recommendations around what CS is
- Examples of where educational leaders can determine appropriate policies and locations in the curriculum to teach computing
- Leadership practices for building principals who can support high-quality teaching
- Understanding equity and access issues associated with computing (Margolis, Goode, & Ryoo, 2014)

What are the *key strategic actions* that a school of education could take to move this work forward

- Computer science educational leaders should participate in conferences and gatherings of educational leadership communities (including superintendents, principals, etc.). Having computer science educators and researchers in these conferences will bring visibility to practicing educational leaders who are often mentors for new administrators.

- Faculty in educational leadership programs should use case studies with problems of practice for CS education implementation. There are numerous examples such as Chicago Public Schools, the New York City public schools, and many smaller city or rural schools with CS.

Focus Area 8:

Creating the Computer Science Education PhD Pipeline

Faculty who focus on CS education are charged with informing research communities and preparing CS faculty who are PhD graduates in CS education. But we have a problem: how do we grow CS education PhDs when we have few CS education faculty? This focus area considers different models for developing a CS education PhD pipeline.

What is the current state of the CS education PhD pipeline?

To inform the development of computer science education, we need researchers in schools of education who will help to answer our core research questions. How do students learn CS, at different grade bands? What are the challenges they face? How do we help them get past those challenges? How are teachers becoming successful CS teachers? What supports their success, and what are the barriers?

Like many discipline-based education research (DBER) fields, computer science education research is starting in CS departments. There are only few computer scientists in post-secondary education who conduct education-specific research. Thus, the computer scientists, often alone, do the computing education research, but this isn't the best situation. Together, disciplinary experts in CS, with education researchers should address questions of CS education from broader points of view.

One challenge right now is that people with a CS PhD struggle to gain legitimacy within the education research community. There is good reason for the skepticism. CS domain experts have to know CS content, but may not have learned about curriculum, pedagogy, assessment, and school policy. They don't necessarily understand the issues of teachers, schools, and students. Researchers with a CS PhD are often focus on education questions related to CS content and what to teach and with what tools whereas researchers with an education PhD are more likely to ask questions regarding student learning and pedagogical practice. There are often different, yet also complementary, theoretical foundations and research methodologies between the two communities.

There are few researchers with CS education PhDs, and right now few or no active formal CS education PhD programs. The opportunities though do exist[12], through individual faculty members rather than established and university listed programs fill

[12] http://faculty.washington.edu/ajko/cer

CSEd at University of Nebraska-Omaha

The University of Nebraska Omaha is a promising example of a school that is hiring computing education researchers in both CS and education academic units and is encouraging collaboration across those units through those faculty. Brian Dorn and Briana Morrison are computing education researchers hired as faculty in Computer Science, and Michelle Friend is a computing education researcher hired as faculty in Teacher Education. They work together on research and on having impact on CS education within the state of Nebraska.

this gap right now. Many of the faculty who are supporting these programs self-identify as "learning scientists," a field that started out as collaborations between education and CS departments[13]. This situation highlights the problem of a clear pathway for either computer scientists or learning scientists to pursue a PhD program which includes the best of both CS content and education content. As a result, we continue to see this division and potential researchers from both disciplines are left without balanced preparation for studying the most difficult questions in CS education.

What does success look like?

A successful CS education PhD pipeline would have entry pathways for either students of CS or education to pursue programs in either departments of CS or schools of education. These programs should take a balanced approach to both methods and the background of the discipline and produce people with PhDs who can cross departmental boundaries. Currently, with a few exceptions, there is a tendency to get specific training in either CS or education, leaving it to the PhD candidate to seek out the balanced experiences needed to become a CS

[13] https://en.wikipedia.org/wiki/Learning_sciences

What would you say to a potential PhD student who asks how to prepare to do CS education research?

At the current time, the best approach is to identify a faculty member in an institution who is studying questions of CS education. The potential student should consider their own background and interests in selecting programs, as the program will be general and need to be supplemented in a way agreed upon by the student and potential advisor to provide either the relevant training in education research or CS content to balance the existing structure. Reading a few of the papers written by potential faculty advisors, or applying for a summer research experience for undergraduates (REU) with a CS education program could be a good start. Andy Ko at the University of Washington maintains an FAQ with advice on how potential PhD students can get started in computing education research.[13]

education researcher.

The two communities also need better shared spaces for development. Often education researchers have difficulty obtaining publication in CS education venues that traditionally serve computer scientists interested in education, and vice versa. The CS education community needs to be thinking about these broader issues as researchers with PhDs in education are trained to ask and explore important questions of education that may involve CS, and can contribute in critical ways to the CS education field.

One successful model are more general computing or informatics PhDs. Several computing education researchers in the US earned a Human-Centered Computing (HCC) PhD at Georgia Tech. The Human-Centered Computing PhD requires students to know learning theory and social science research methods. Informatics programs, such as at University of Washington, Seattle, and Drexel University, have produced successful computing education researchers who advance our knowledge of how to teach CS.

Once the pathways are created, researchers who earn PhDs in Computer Science Education will need positions that will allow them to continue to investigate research questions at the intersection of the two domains. Currently, CS education researchers need a discipline-specific lens in order to obtain a faculty position where they can explore questions of CS education. Some faculty exist in schools of education and teach classes in educational technology, or basic methods. Others exist in schools of CS and teach CS content courses. With few CS education programs to prepare teachers, as mentioned in previous sections, there are few opportunities for faculty to teach courses in CS education. Additionally, CS education researchers have found positions in local education agencies (school districts or state departments of education), research firms such as AIR and others, or CS focused companies such as Google who have made CS education a priority.

We do want to have CS education researchers coming from and working in traditional education programs. We also want faculty who know CS education research building teacher development programs, working with education graduate students, and advancing a computing education research agenda from an education perspective, not just a computing perspective. There are challenges to growing CS education in school of education.

What is the gap between success and where we are now?

A major issue is the identification of faculty with balanced backgrounds of CS and education disciplinary knowledge to lead such PhD programs in schools of education. Identifying current CS faculty who do

[14] http://faculty.washingtonedu/ajko/cer

education research may not have publication records or CVs that are appropriate for hiring within a school of education. Once hired, there will also be challenges for such faculty regarding promotion. For example, there is a process for how Assistant Professors become tenured Associate Professors. Within the CS field, conference proceedings are rated higher while within education schools, journal articles are rated higher because the education community doesn't have a history of published conference proceedings as technical fields like CS do. For a professional who exists in each domain, there is a need to explain to promotion committees why either aspect—conference or articles depending on the committee—should be valued in the same way as any other accomplishment that contributes to the highest value. There is an innate difficulty in bridging the two worlds of CS research with a focus on education and education research with a focus on CS because the promotion committees and outside reviewers might not always understand the competing priorities due to the differences in the disciplinary domains.

The structure of academic institutions right now make this a difficult challenge. It is not impossible and there are only some schools that understand the importance of computing education and the need to get beyond academic silos to make progress.

What are the *key strategic actions* that a school of education could take to move this work forward

- ☆ Schools of education and schools of CS should work together to define opportunities for interdisciplinary coursework for interested CS education PhD candidates.
- ☆ Current CS education faculty should propose and seek funding for workshops for leaders of schools of education to promote understanding of quality CS education publications and CVs for faculty candidates.
- ☆ If there are current faculty engaged in CS education research, the school should seek to highlight opportunities for PhD candidates to work with these faculty. If not, schools of education should consider hiring candidates with CS education research experience to produce a well-rounded faculty.

References

Borman, G. D., & Dowling, N. M. (2008). Teacher attrition and retention: A meta-analytic and narrative review of the research. Review of Educational Research, 78(3), 367-409.

Century, J., Lach, M., King, H., Rand, S., Heppner, C., Franke, B., & Westrick, J. (2013). Building an operating system for computer science. Chicago, IL: CEMSE, University of Chicago with UEI. Retrieved from http://outlier.uchicago.edu/computerscience/OS4CS/

Emechebe, O. (2015, September 25). Professor wants to double the number of computer science teachers in Wisconsin. Retrieved from http://host.madison.com/ct/news/local/education/professor-wants-to-double-the-number-f-computer-science-teachers/article_e2183c96-ab3d-5ce9-a5ba-85fb79fe97c1.html

Ericson, B., & Guzdial, M. (2014, March). Measuring demographics and performance in computer science education at a nationwide scale using AP CS data. In Proceedings of the 45th ACM technical symposium on computer science education (pp. 217-222). ACM.

Flannery, M. (2016, March 23). Survey: Number of future teachers reaches all-time low. Retrieved from http://neatoday.org/2016/03/15/future-teachers-at-all-time-low/

Goode, J. (2007). If you build teachers, will students come? The role of teachers in broadening computer science learning for urban youth. Journal of Educational Computing Research, 36(1), 65-88.

Goode, J., Margolis, J., & Chapman, G. (2014). Curriculum is not enough: The educational theory and research foundation of the exploring computer science professional development model. In Proceedings of the 45th ACM Technical Symposium on Computer Science Education, 493-498.

Guzdial, M. (2014). Preparing teachers is different than preparing software developers: [WiPSCE'14 keynote]. In Proceedings of the 9th Workshop in Primary and Secondary Computing Education (WiPSCE '14). ACM, New York, NY, 1-1. doi: 10.1145/2670757.2676866

Guzdial, M. (2015). Learner-Centered Design of Computing Education: Research on Computing for Everyone. Willston, VT: Morgan & Claypool.

Guzdial, M. (2016). Bringing computer science to US schools, state by state. Communications of the ACM, 59(5), 24-25.

Hubwieser, P., Armoni, M., Giannakos, M. N., & Mittermeir, R. T. (2014). Perspectives and visions of computer science education in primary and secondary (K-12) Schools. ACM Transactions on Computing Education (TOCE), 14(2), 7.

Hubwieser, P., Giannakos, M. N., Berges, M., Brinda, T., Diethelm, I., Magenheim, J., ... & Jasute,E. (2015, July). A global snapshot of computer science education in K-12 schools. In Proceedings of the 2015 ITiCSE on Working Group Reports (pp. 65-83).
New York, NY: ACM.

Kafai, Y. B. & Burke, Q. (2014. Connected code: Why children need to learn programming. Cambridge, MA: The MIT Press.

Lang, K., Galanos, R., Goode, J., Seehorn, D., Trees, F., Phillips, P. and Stephenson, C. (2013). Bugs in the system: Computer science teacher certification in the U.S. NY: Computer Science Teachers Association.

Margolis, J., Estrella, R., Goode, J. & Holme, J. & Nao, K. (2017). Towards the Shallow End (revised edition). Cambridge, MA: The MIT Press.

Margolis, J., Goode, J., Chapman, G., & Ryoo, J. J. (2014). That classroom 'magic'. Communications of the ACM, 57(7), 31-33.

Margolis, J., Goode, J., & Ryoo, J. J. (2015). Democratizing Computer Science. Educational Leadership, 72(4), 48-53.

McLeod, S. H., & Soven, M. (1992). Writing across the curriculum. Writing Across the Curriculum, Newbury Park, CA: Sage.

Next Generation Science Standards (2013). Next Generation Science Standards: For States, By States. Washington, DC: The National Academies Press.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc.

Parker, M. C., & DeLyser, L. A. (2017, March). Concepts and Practices: Designing and Developing A Modern K-12 CS Framework. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (pp. 453-458). ACM.

Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2018). K-8 learning trajectories derived from research literature: sequence, repetition, conditionals. ACM Inroads, 9(1), 46-55.

Sadler, P. M., Sonnert, G., Coyle, H. P., Cook-Smith, N., & Miller, J. L. (2013). The Influence of Teachers' Knowledge on Student Learning in Middle School Physical Science Classrooms. American Educational Research Journal, 50(5), 1020-1049.

Schanzer, E., Fisler, K., Krishnamurthi, S. (2018). Assessing Bootstrap: Algebra Students on Scaffolded and Unscaffolded Word Problems. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). ACM, New York, NY, USA, 8-13. doi: 10.1145/3159450.3159498

Seehorn, D., & Clayborn, L. (2017, March). CSTA K-12 CS Standards for All. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (pp. 730-730). ACM.

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. Educational Researcher, 15,4-14.

Thomas, L. G., & Knezek, D. G. (2008). Information, communications, and educational technology standards for students, teachers, and school leaders. In International handbook of information technology in primary and secondary education (pp. 333-348). Springer, Boston, MA.

Wang, J., Hong, H., Ravitz, J., & Hejazi Moghadam, S. (2016, February). Landscape of K-12 computer science education in the US: Perceptions, access, and barriers. In Proceedings of the 47th ACM Technical Symposium on Computing Science Education (pp. 645-650). New York, NY: ACM.

Yadav, A., Korb, J. T. (2012). Learning to teach computer science: The need for a methods course. Communications of the Association for Computing Machinery, 55(11), 31-33.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. TechTrends, 60(6), 565-568.

Appendix

APPENDIX A:

Computer Science Courses and other Resources

AP Computer Science Principles A: a college-level semester-long course taught in many high schools in preparation for the AP exam

(https://apstudent.collegeboard.org/apcourse/ap-computer-science-a)

Beauty and Joy of Computing: a College Board approved AP level course that uses the SNAP programming language.

(https://bjc.berkeley.edu)

Code.org: Computer science programs for the middle school and elementary school levels that provide a broad introduction in various computer science topics.

(https://code.org/educate/curriculum/middle-school)

(https://code.org/educate/curriculum/elementary-school)

Creative Computing by ScratchEd Team HGSE: A guide that outlines a series of introductory programming projects using Scratch programming language.

(http://scratched.gse.harvard.edu/guide/)

CS Principles: a College Board approved AP level course that uses uses App Lab and Javascript programming languages.

(https://studio.code.org/courses/csp)

Exploring Computer Science: A year-long high-school level course that provides overview on various computer science topics such as human-computer interaction, problem solving, Scratch programming, data science, web design and robotics.

(http://www.exploringcs.org)

MakeCodePlay: A guide that describes the design of games and controller activities on and off the screen to introduce students to various computational thinking concepts and practices using Scratch programming language.

(http://www.yasminkafai.com/MCPs)

Mobile CS Principles: a College Board approved AP level course that uses uses the App Inventor for Android programming language

(https://course.mobilecsp.org)

Programming, Games, and Apps Society: Provides various modules that introduce students to app design and related issues.

(http://www.gavirtuallearning.org)

Project Lead the Way: a College Board approved AP level course that uses App Inventor and Java programming languages.

(https://www.pltw.org)

Scalable Game Design: Provides a sequence of game design tasks that introduce students to key computational concepts and practices.

(https://sgd.cs.colorado.edu/wiki/Scalable_Game_Design_wiki)

Teaching of Programming in Scratch: A course offered by Lehman College, a CUNY school, which offers computer science credit for students in the school of education.

(link)

UTeach CS Principles: a College Board approved AP level course that uses uses Scratch processing programming language.

(cs.uteach.utexas.edu)

APPENDIX B:

List of Workshop Attendees

June Ahn, Ph.D., Associate Professor, New York University

Connor Bain, Ph.D. Candidate, Northwestern University

Matthew Berland, Ph.D., Associate Professor, University of Wisconsin-Madison

Sarah Bonner, Ph.D., Associate Professor, Hunter College

Karen Brennan, Ph.D., Associate Professor, Harvard Graduate School of Education

Jan Cuny, Ph.D., Program Director for Computing Education, National Science Foundation

Leigh Ann DeLyser, Ph.D., Co-Founder and Managing Partner, CSforALL

Barbara Ericson, Ph.D., Director of Computing Outreach, Georgia Institute of Technology College of Computing

Robert Floden, Ph.D., Dean, College of Education, Michigan State University

Diana Franklin, Ph.D., Associate Professor, University of Chicago

Michelle Friend, Ph.D., Assistant Professor, University of Nebraska, Omaha

Joanna Goode, Ph.D., Associate Professor, University of Oregon College of Education

Mark Guzdial, Ph.D., Professor, Georgia Institute of Technology College of Computing

Nathan Holbert, Ph.D., Assistant Professor, Teachers College, Columbia University

Hai Hong, M.S.Ed., Program Manager, K-12 Education Outreach, Google

Kimberly Hughes, M.Ed., Director, UTeach Institute

Yasmin B. Kafai, Ed.D., Professor, Graduate School of Education, University of Pennsylvania

Leslie Keller, MBA, Computer Science Education Program Manager, North Carolina State University

Colleen Lewis, Ph.D., Assistant Professor, Harvey Mudd College

Mark Nelson, Ph.D., Innovation Consultant

Lijun Ni, Ph.D., Professor, University at Albany, SUNY

Anne Ottenbreit-Leftwich, Ph.D., Associate Professor, Indiana University

Anthony Petrosino, Ph.D., Associate Professor, University of Texas at Austin

Wesley Pitts, Ph.D., Professor, Lehman College

David Pugalee, Ph.D., Director, Center for STEM Education, University of North Carolina-Charlotte

Gaoyin Qian, Ph.D., School of Education Associate Dean, Lehman College

Debra Richardson, Ph.D., Professor/Founding Dean, University of California, Irvine

Deborah Seehorn, Curriculum Committee Chair, Computer Science Teachers Association

Beth Simon, Ph.D., Associate Teaching Professor, University of California, San Diego

Nicki Washington, Ph.D., Professor, Winthrop University

David Weintrop, Ph.D., Assistant Professor, University of Maryland

Joseph Wilson, Ph.D., Senior Education Consultant, American Institutes for Research

Aman Yadav, Ph.D., Associate Professor, Michigan State University