The Need for

To keep up with career opportunities, young people will need to understand computer science. So why do so few schools offer it?

Jane Margolis, Joanna Goode, and David Bernier

Computer science is driving innovation—from the sciences to the arts. A look at any major news source reveals the effect of computer science and technology on the global economy. Yet in the majority of U.S. high schools, especially in low-income and underserved communities, students rarely have access to computer science classes.

Narrowly defined, computer science is based on a core set of problem-solving concepts; it has been defined as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society" (Association of Computing Machinery, 2003). But it doesn't stop there. Computer science is a lens and an entry into skills in critical and logical thinking that apply across all disciplines, including writing and the humanities (Carey, 2010).

The percentage of high schools with rigorous computer science courses fell from 40 percent in 2005 to 27 percent in 2009.

Although policymakers express the need for more computer science expertise, learning opportunities in secondary schools are actually shrinking. The Association of Computing Machinery (the national professional organization for computer scientists) reports that the percentage of high schools with rigorous computer science courses fell from 40 percent in 2005 to 27 percent in 2009 (Computer Science Education Week, n.d.).

Where computer science courses exist, too often they are not engaging students with the problem solving, logical thinking, and creativity of computer *science* (Computer Science Teachers Association [CSTA], 2005; Margolis, Estrella, Goode, Holme, & Nao, 2008; Wing, 2008). In too many schools, the computing curriculum is defined by basic computer literacy and rudimentary computing skills, such as typing, word processing, Internet searching, and working with spreadsheets.

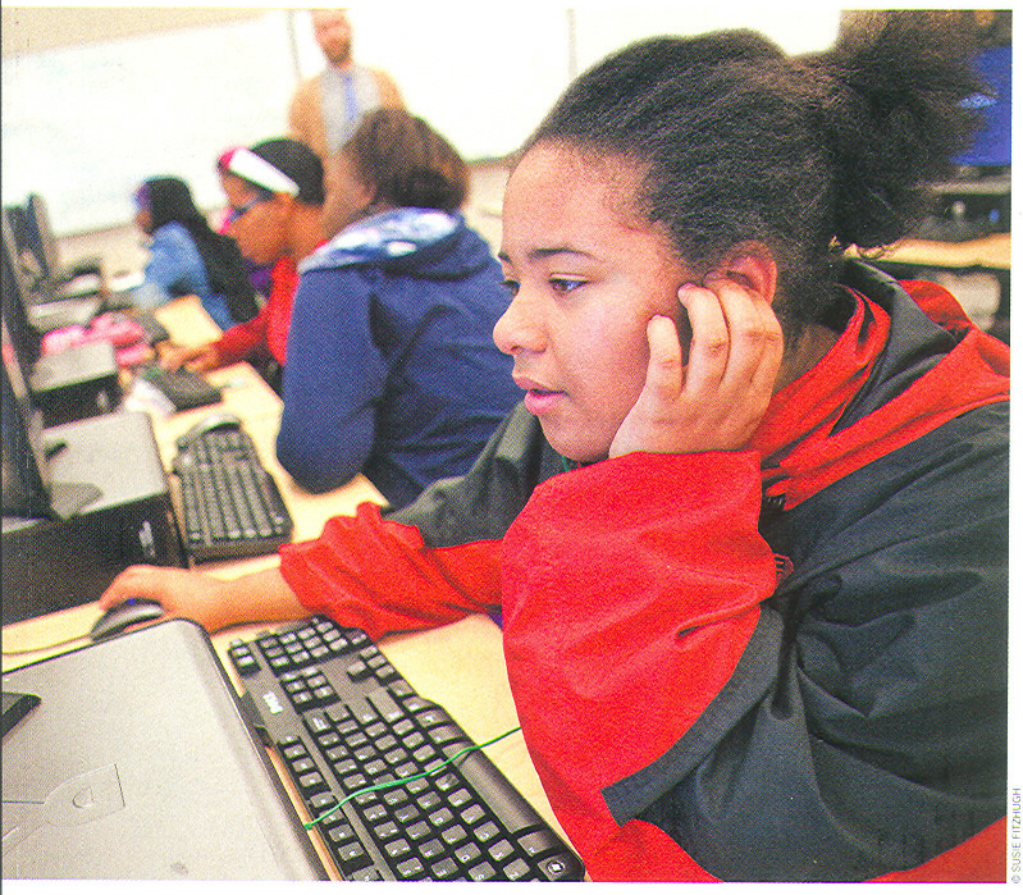
Compounding this situation is a computer science teacher shortage. Because computer courses are not generally designated as academic courses (but rather as technical arts or vocational classes), there is no official computer science teacher certification or established course in how to teach the discipline (CSTA, 2008; Margolis et al., 2008). Computer science teachers are often teaching out of subject and lacking a professional learning community.

As a result, across the United States, only a narrow, largely homogenous band of students—frequently those from families able to provide computers, Internet access, robotic kits, a plethora of software, and parental knowledge—are introduced to computer science. Many others, who lack high-quality schooling opportunities and substantial family resources, are relegated to the shallow end of computing skills.

Access and Equity

From 2000 to 2003, our research team addressed this equity problem in computer science education and studied why so few black and Latino students and girls were learning computer science in the Los Angeles Unified School District, the second largest district in the United States. At that time, 8 percent of students taking the Advanced Placement

Computer Science



© SUSIE FITZHUGH

Computer Science A Exam in California were Latino or Latina (of a high school population that was 41 percent Latino/ Latina) and only 1 percent were black (of a high school population that was 8 percent black). Further, only 18 percent were female (National Center for Education Statistics, 2004). Our findings, which we described in our book *Stuck in the Shallow End: Education, Race, and Computing* (MIT Press, 2008), revealed that schools with high numbers of students of color commonly offer only the lowest-level introductory computer classes. College-preparatory computer science courses and teachers with sufficient background to teach them are largely missing.

In response to these findings, in 2004 we formed the Computer Science Equity Alliance, a partnership between University of California at Los Angeles (UCLA) researchers and Los Angeles Unified School District (LAUSD) officials. Our goal was to build the district's capacity to give more K–12 students access to engaging, rigorous computer science learning. The alliance's efforts resulted in the doubling of advanced placement (AP) computer science courses in Los Angeles schools and a dramatic increase in the number of Latino, black, and female students taking these classes (Goode, 2008).

At the same time, we discovered there was a need for a more accessible and engaging introductory college-preparatory computer science curriculum, one that was committed to broad student participation. In response, our partnership developed an innovative college-preparatory high school computer science course, Exploring Computer Science, that is a precursor to AP Computer Science A. The curriculum for the course is now available at www.exploringcs.org. This yearlong course is now offered in more than 16 LAUSD high schools. During the past school year, 921 students enrolled in the course; 644 of these students are Latino, 94 are black, and 342 are girls of all ethnicities.

To support Exploring Computer Science teachers, we have implemented a professional development and coaching program and a professional learning community for teachers. We've also created systematic channels so this course will expand to more schools and continue to be supported.

Although many aspects of our work are specific to Los Angeles, schools around the United States that hope to offer more—and more rigorous and engaging—computer science courses can gain insight from what we found works in building a school's capacity to increase computer science learning opportunities.

Broadening Engagement with Computer Science

One barrier to increasing the number of students taking courses in computer science is that too many students (and teachers) identify success in computer science with a narrow segment of the student population. This is ironic,

PHOTO BY JEAN RYOO

Exploring Computer Science students work together to assemble a robot.

considering most teenagers' fascination with computers and social media. Yet, because computer scientists in the United States are predominantly white or Asian males, many students assume that computer science is "not for me." We pushed hard—and successfully—to overcome this barrier and made a commitment to have Exploring Computer Science taught in schools with high numbers of students of color. We introduced assignments that engage a broad segment of students and connect to their lives and communities. Clearly, we can't accurately gauge students' interest, motivation, and achievement in computer science unless we provide them with equal access to meaningful courses.

Exploring Computer Science has six instructional units: (1) Human-Computer Interaction; (2) Problem Solving; (3) Web Design; (4) Introduction to Programming (which uses the programming language Scratch); (5) Robotics; and (6) Computing Applications (data analysis). Rather than focusing on acquiring a specific computer programming language and its syntax (the focus of the current AP Computer Science A course), the overarching goal of this course is to engender creativity and promote exploration around the foundational concepts of computer science. The course helps students see the capabilities of humans working with computers to analyze and solve problems and conduct real-world tasks.

In the context of discussing issues that affect their lives, students learn about abstraction, algorithms, and other computer science concepts while creating websites; designing animation programs (thus learning the fundamental practices of programming); and building and programming robots. They learn that computer science is about people, society, and community—not just machines and systems—and they explore how the discipline affects the world today.

Mobilizing for Learning

This year, we are embarking on a new project involving our Exploring Computer Science students called Mobilizing for Innovative Computer Science Teaching and Learning, in collaboration with the UCLA Center for Embedded Networked Sensing (CENS) and the Computer Science Teachers Association.[1] The Mobilize project builds on teenagers' engagement with mobile technology. At the heart of the project is the CENS participatory sensing system—an innovative method of data collection and analysis in which individuals use mobile phones to systematically collect and interpret data about issues important to them and their communities. These "data campaigns" provide rich opportunities to learn about what constitutes robust data and how to best analyze and represent data in the context of math, science, or computer science courses; they also engage students' sense of civic involvement.

Mobilize will create hands-on, inquiry-based curricular units that use the participatory sensing system—and accompanying professional development—for high school computer science, mathematics, and science classes. Our team will first introduce these units and projects in Exploring Computer Science classrooms. This will be followed by implementation by interdisciplinary teams of teachers from Exploring Computer Science, mathematics, and life and physical science classes, joined by social science students and teachers working on Mobilize projects in Los Angeles schools.

One goal of Mobilize is to strengthen computer science instruction throughout our education system. The number of topics for which teachers might create participatory sensing projects is vast; the following is a far from exhaustive list of sample data campaigns:

- *Transportation*: Students map their travel patterns and analyze them for ways they might reduce their carbon footprints.
- *Recycling*: Students document the availability and actual usage of recycling bins around campus or in their

neighborhoods, analyze existing options to make recycling collection more widespread and successful, and estimate possible savings if these changes were implemented.

■ *Daily habits.* Students select a personal habit, such as nail biting, that they want to manage better. They program queries into their phone to signal them several times a day to report on whether they are engaging in the problem behavior at that moment. Students then use statistical techniques to look for correlations between doing the problem behavior and other factors in their lives.

Having a beyond-basic knowledge of computers connects a student to a grid of opportunities.

Breaking Stereotypes

Given the low numbers of girls and students of color studying computer science across the United States (Zweben, 2006), we've made a point of presenting diverse role models. This past spring, our partnership brought more than 500 Exploring Computer Science students, teachers, parents, and staff members from Los Angeles schools to UCLA. The highlight of the day was a presentation from the visiting all-female African American Spelman College Robotics Team. The Spelbots shared their personal stories while modeling how they programmed a Nao robot to use its motors, cameras, and sensors to respond to voice commands, play soccer, and dance to a popular song. Students commented about how much they learned about robotics and how much they appreciated seeing young black women involved in this field. One student noted, "Spelbots team members contradicted common stereotypes, [which] motivated me to want to learn computer science no matter what others say."

Through the Exploring Computer Science course, we have found that teenagers who never would have thought of themselves as computer types became fascinated by computer science. Juan, a nonnative English speaker, got so hooked on the programming language Scratch that now when he plays video games, he notices programming elements—"oh, that's a forever loop." Juan wants to take Advanced Placement Computer Science, and the effect of taking Exploring Computer Science has carried over to other disciplines. "If I can do ECS, then I can do chemistry too," he said.

Nancy, a Latina high school junior, had never taken a computer science class; her friends teased her about her "Neanderthal" level of skill with computers. After taking Exploring Computer Science, Nancy wants to take AP Computer Science next year. Our surveys showed that, overall, students' interest and engagement with computer science significantly increased, and teachers indicated that students were more prepared for advanced computer sciences classes after taking the course.

Helping Teachers Make the Shift

A crucial part of building capacity for computer science instruction is providing professional development and support for teachers. In addition to an innovative curriculum, teachers must have extensive content knowledge and practice with pedagogy that engages students. Because many computer teachers are used to more traditional instruction methods (and are teaching out of subject), a big shift is required to teach a hands-on project-based curriculum like Exploring Computer Science.

For the Los Angeles teachers, we've developed a summer institute focused on course content and pedagogy. Inquiry-focused instruction is modeled through role-playing, pair and small-group collaboration, structured tinkering, simulations, and the promotion of the belief that there is more than one way to solve a problem. During the academic year, we hold follow-up Saturday sessions. This ongoing professional development gives teachers a regular chance to come together with fellow teachers and university educators to reflect on their teaching experiences, prepare for the breadth of content involved in teaching computer science, and build a learning community.

We discovered that in-classroom follow-up is also necessary, so we created a coaching program. Coaches and teachers design a collaborative plan that identifies goals for how teachers want to improve, success indicators, and strategies to achieve those goals—such as coteaching and videotaping one's own practice. We currently have two full-time computer science coaches serving about eight schools each. The frequency of their visits varies according to teacher interest and availability, ranging from weekly with some teachers to quarterly with others. Coaches model lessons as well as guide teachers in answering student questions that they had struggled to answer.

Taking It National

Exploring Computer Science is now expanding throughout Los Angeles and California; the Oakland and San Jose school districts plan to offer it soon. Across the United States, efforts to increase computer science learning opportunities at the K–12 level are also being spearheaded by the National Science Foundation's (NSF) Broadening Participation in Computing program, the Computer Science Teachers Association, the Association of Computing Machinery, and the National Center for Women in Technology.

Teenagers who never would have thought of themselves as computer types became fascinated by computer science.

The NSF Computer Science 10K Project aims to develop an entirely new high school computing curriculum and to get that curriculum taught by 10,000 well-prepared teachers in 10,000 schools by 2015. The curriculum will have an introductory course (like Exploring Computer Science) followed by a new advanced placement course. For efforts like these to flourish, states and school districts will need to advocate similar initiatives and implement such models locally in committed, creative ways.

Because computer science drives innovation, discovery, and even political participation, learning about computer science is important and benefits all students. Computer science instruction teaches students how to be innovative and solve problems. Further, because of students' fascination with technology, it has the potential to reach many students who are not successfully engaged by the rest of the academic core. And in today's world, having a beyond-basic knowledge of computers connects a student to a grid of opportunities. Being introduced to engaging computer science in high school can propel students who otherwise would never have realized they enjoyed high-level computing into the field.

Too often, educators view the absence of black, Latino, and female learners in computer science as indicative of lack of interest, motivation, or ability. In our research, we found that an interaction between school structural issues (lack of courses and teachers) and belief systems (low expectations) resulted in disparities in access to computer science along racial, socioeconomic, and gender lines (Margolis et al., 2008). The dearth of computer science learning opportunities was then justified by stereotypes and assumptions about which type of students are interested in or capable of doing computer science—and which are not.

Forecast for the Future

As we enter our third year of supporting Exploring Computer Science teachers and students—at this time of budget cuts and a national education crisis—we recognize how fragile this work can be. Our reform model calls for curricular and pedagogical changes, a teacher learning community, and policy changes that will grant computer science academic credit and establish a computer science teacher certification pathway. We believe that a K–12/university partnership with a social justice perspective—addressing structural factors, belief systems, and policy changes—is essential for sustaining these reforms.

Although some of the particulars are unique to Los Angeles schools, we hope our description of this reform model will provide guidance for other districts, schools, or district-university partnerships that hope to broaden participation in computing. Career and education opportunities are being transformed by technology and computer science. Broadening computer science learning beyond a narrow band of privileged students must be a part of our education and equity agenda. EL

[1]This project is supported by a National Science Foundation Math Science Partnership. (See www.exploringcs.org.)

References

Association of Computing Machinery K–12 Task Force Curriculum Committee. (2003). *A model curriculum for K–12 computer science.* Retrieved from Computer Science Teachers Association at www.csta.acm.org/Curriculum/sub/ACMK12CSModel.html

Carey, K. (2010, November 7). Decoding the value of computer science [online article]. *Chronicle of Higher Education.* Retrieved from http://chronicle.com/article/Decoding-the-Value-of-Computer/125266/

Computer Science Education Week. (n.d.). *Data/computer science facts.* Retrieved from www.csedweek.org/resources/datacs-facts

Computer Science Teachers Association. (2005). *The new educational imperative: Improving high school computer science education* (Final Report of the CSTA Curriculum Improvement Task Force). New York: Association for Computing Machinery.

Computer Science Teachers Association. (2008). *Ensuring exemplary teaching in an essential discipline: Addressing the crisis in computer science teacher certification* (Final Report of the CSTA Teacher Certification Task Force). New York: Association for Computing Machinery.

Goode, J. (2008). Reprogramming college preparatory computer science. *Communications of the ACM, 51*(11), 31–33.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2008). *Stuck in the shallow end: Education, race, and computing.* Cambridge, MA: MIT Press.

National Center for Education Statistics. (2004). *Common core data.* Washington, DC: Author.

Wing, J. (2008). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Zweben, S. (2006, May). 2004–2005 Taulbee survey. *Computing Research News, 18*(3), 7–17. Retrieved from http://archive.cra.org/statistics/survey/05/05.pdf

Jane Margolis is senior researcher at the UCLA Graduate School of Education and Information Studies and coauthor (with Joanna Goode) of *Stuck in the Shallow End: Education, Race, and Computing* (MIT, 2008); margolis@ucla.edu. **Joanna Goode** is assistant professor in the Department of Education Studies at the University of Oregon, Eugene; goodej@uoregon.edu. **David Bernier** is director of the Computer Science Project at Center X in the UCLA Graduate School of Education and Information Studies; dgbernier@gmail.com.