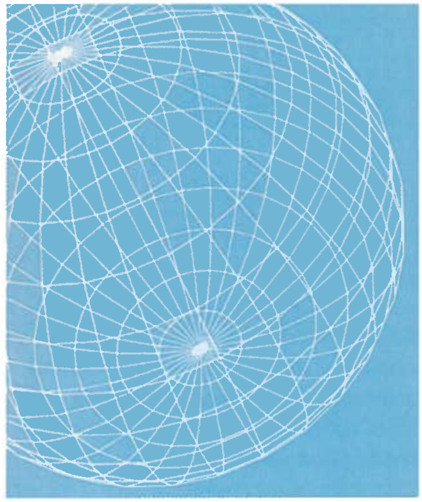
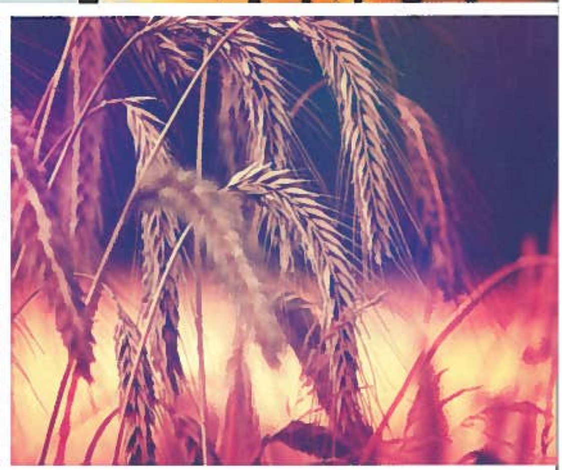
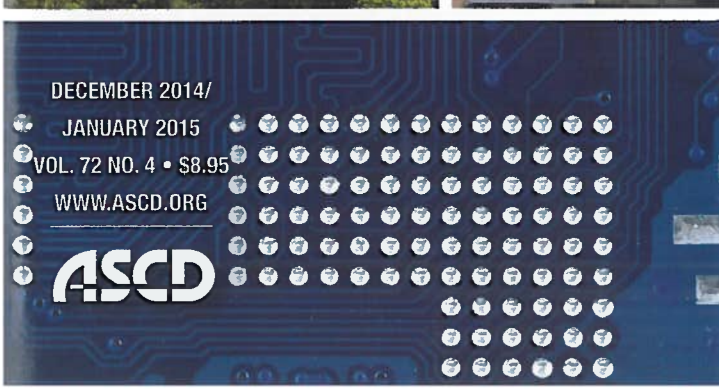
EL
EDUCATIONAL LEADERSHIP



STEM FOR ALL





DECEMBER 2014/
JANUARY 2015
VOL. 72 NO. 4 • $8.95
WWW.ASCD.ORG
ASCD

Democratizing Computer Science

Why we need to open computer science courses for more students.

Jane Margolis, Joanna Goode, and Jean J. Ryoo

A high school computer science class begins with students divided into groups, writing instructions for making a peanut butter and jelly sandwich. One group of "programmers" goes first, reading instructions to the "computer" (fellow students) for making the sandwich. But a "programming error" occurs: The computer cannot operate because the "algorithm" (instructions) did not specify taking off the lid of the peanut butter as an initial step. Student laughter ensues. Computers are dumb!

The process begins again, with a revised sequence of instructions. But the computer spreads the peanut butter on the table because the algorithm did not specify to spread it on the bread. Whoops and hollers everywhere: Computer science can be fun! All students are engaged.

So begins an early lesson in Exploring Computer Science (ECS), an introductory high school computer science course supported by the National Science Foundation (www.exploringcs.org). ECS's mission is to demystify computer science by introducing students to the fundamental concepts of the field. Students learn about human-computer interaction, problem solving, web design, programming, data analysis and computing, and robotics. Throughout all units, students learn to use computing skills in ways that are meaningful for themselves and their communities.

An Equity Imperative

Computer science is generally identified with a narrow stratum of our student population, but all students today need to learn about it because computers are changing how we communicate, innovate, work, play, and think—regardless of career choice. Computer science relies on problem solving and computational practices that are important for everyone. These include using abstractions; automating; creating algorithms; collecting and analyzing data; implementing, testing, and debugging designs; and engaging in creative, critical, and innovative thinking.

Despite its growing importance, computer science is on the sidelines in our schools. The primary

PHOTO COURTESY OF JEAN J. RYOO

PHOTO COURTESY OF JOHN LANDA

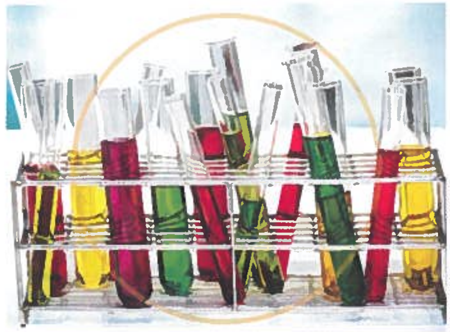
Students develop interest and ability in computer science **when they have opportunities** to engage with it.

college-preparatory course is Advanced Placement (AP) Computer Science, which focuses on programming and fails to attract many students. Computer science is rarely part of the academic core. Further, only a handful of states offer a computer science teacher preparation program that leads to certification.

Computer science is a subject that many students (and adults!) find intimidating. It is stereotypically associated with super-smart, geeky, white or Asian males, many of whom benefited from having resources at home to foster their interest. The statistics of who is (and is not) studying computer science reveal the depth of the problem.

For example, 53 percent of K–12 public school students in California schools are Latino; yet only 8 percent of the nearly 5,000 California students taking the AP Computer Science exam last year were Latino. Six percent of California students are black, but only 1 percent of students taking the exam were black. And only 22 percent of exam takers were female (California Department of Education, 2012; College Board, 2013).

The good news is that a growing number of school districts are working to bring computer science into schools. Exploring Computer Science was first developed in 2008 in partnership with Los Angeles Unified School District as a response to research detailed in *Stuck in the Shallow End: Education, Race, and Computing* (Margolis, Estrella, Goode, Holme, & Nao, 2008). Research conducted in Los Angeles high schools found that schools with high numbers of low-income students of color were offering courses labeled as computer science that included little more than keyboarding and other rudimentary skills. To address this issue and broaden participation in computing, the ECS curriculum was written as an introductory computer science course designed to engage all students in the subject.


COWARDLION/SHUTTERSTOCK

Science is often misrepresented as "the body of knowledge acquired by performing replicated controlled experiments in the laboratory." Actually, science is something broader: the acquisition of reliable knowledge about the world.

—Jared Diamond
From *Collapse: How Societies Choose to Fail or Succeed*

Making Computing Relevant

Exploring Computer Science departs substantially from traditional computer science classes focused on programming in which a student predominantly works alone at a computer. Instead, the ECS curriculum is structured around collaborative inquiry practices through which students together explore, design investigations, think critically, test solutions, and solve real problems. ECS classrooms are noisy and active places that encourage exploration, autonomy, persistence, risk taking, and creativity by insisting that there are multiple solutions to a problem and giving students opportunities to seek those solutions for themselves. Sometimes, student work doesn't even require a computer, such as when students learn to think like programmers by writing instructions for making peanut butter and jelly sandwiches.

For traditionally underrepresented students, engagement and learning are deepened when computer science practices are presented in meaningful ways that blend students' social worlds with the world of science (Barton & Tan, 2010). Consider these activities that draw on students' local and cultural knowledge:

- In the first ECS unit on human-computer interaction, students learn about Internet searching by conducting scavenger hunts for data about the demographics, income level, cultural assets, people, and educational opportunities in their communities.
- In the problem-solving unit, students work with Ron Eglash's Culturally Situated Design Tools (http://csdt.rpi.edu), software that helps students see the mathematical and computing principles behind cultural artifacts. In one example, students learn about the history of cornrow braiding and how cornrow designs involve transformational geometry.
- In the web design unit, students learn to use HTML and CSS (cascading style sheets) to create websites on any topic of their choosing, such as an ethical dilemma, their family tree, a possible future career, or worldwide or community problems.
- In the introduction to programming unit, students use the Scratch programming language to create a game or an animated story about an issue of concern.
- In the data analysis and computing unit, students collect and combine data about their snacking behavior and learn to analyze the information and compare it with large data sources.
- In the robotics unit, students program robots to move through mazes or dance to students' favorite songs.

Each unit concludes with a project that connects students' worlds to computer science concepts. For example, in the problem-solving unit, students might connect their knowledge of problem solving, data collection, and minimum spanning trees (subgraphs that connect points on a larger graph) to create the shortest and least expensive route for showing tourists their favorite places in their neighborhood.

Preparing Teachers

In addition to a curriculum, the ECS program includes a professional development program for teachers. It's not easy for teachers to transition from content delivery methods of teaching to inquiry-based instruction. Professional development that builds and supports an ongoing teacher learning community is key. The ECS professional development program spans two years with a combination of two summer weeklong institutes and quarterly Saturday workshops, as well as in-classroom coaching, informal teacher meetings, classroom visits, conferences, and more.

The professional development mirrors how students learn in ECS classrooms. During the summer institute, teachers are immersed in inquiry-based learning as they work in small groups to plan, coteach, and reflect on introductory lessons for fellow teachers who participate as "students." After each lesson, the whole group discusses the implications for teaching these lessons to diverse students: When in the lesson were they the most engaged? The least? Who was never engaged? Why? What strategies contributed to engagement?


PHOTO COURTESY OF JEAN J. RYOO

The learning continues throughout the school year with monthly professional learning community meetings in which teachers discuss their experiences teaching ECS. The program also includes in-classroom coaching in which coaches who have experience teaching computer science and ECS classes help teachers reflect on how they're implementing the curriculum.

Through teaching ECS, teachers gain an increased appreciation for students' abilities to engage with computer science (Goode, Margolis, & Chapman, 2014). One teacher commented that "I don't need to completely understand each topic in order to teach it—this lets some of my students become the masters and then they can teach the other students." Teachers acquire new understandings of how to facilitate student inquiry, active learning, and student ownership of their learning and of how to draw out their students' investigative abilities and creativity.

In a field with strong biases and stereotypes, teachers' new appreciation for students' interest and capacity are essential to broadening participation in computing. Computers don't create cultural change; teachers do. And for this to occur, teachers need the space to think big and be bold about their craft. But it is not just teachers who must change. It is the entire school culture.

Broadening Participation

One of the missions of Exploring Computer Science is to demonstrate that students develop interest and ability in computer science when they have opportunities to engage with it. Keeping the content rigorous yet accessible, engaging, and relevant to diverse students is crucial for broadening participation in computing.

Enrollment patterns of Los Angeles Unified School District (LAUSD) students in ECS reveal high participation rates that closely mirror district demographics. In 2013–14, more than 2,500 LAUSD students enrolled in ECS; 73 percent were Latino, 11 percent black, 7 percent Asian, 8 percent white, and 46 percent female. These ECS demographics are a dramatic contrast to the severe underrepresentation of nonwhite and female students in computer science generally, as indicated by the AP Computer Science statistics (College Board, 2013). ECS has also grown to be a national program with 13 regional programs, including partnerships in the largest three school districts in the United States: Los Angeles, Chicago, and New York City.

Los Angeles student data collected through surveys and interviews show a promising increase in student interest, engagement, confidence, and persistence with learning computer science. There have been statistically significant increases in self-perceived knowledge across all ECS topics, with the largest increases associated with robotics and programming. When reflecting on what she considered a "nerve wracking" process of creating a game in the programming language Scratch, one student said,

> Now I have kind of an idea of how things work, how the cartoonists do stuff and, yeah, it's been awesome. . . . After all of the frustration, it leaves you a satisfying feeling because you're like, "Yes, I did it."

After taking ECS, when students are

asked to describe how computer scientists think, both male and female students are significantly less likely to use words that focus exclusively on intelligence (as they did in the pre-class surveys) and are significantly more likely to use words related to computational practice (such as *analyze*, *program*, or *problem solve*). Female students show a significant increase in growth mind-sets (Dweck, 2007) as they come to view computer science ability more in terms of experience and hard work than innate aptitude. This shift in beliefs is essential to drawing more students into computer science.

PHOTO COURTESY OF JOHN LANDA

ECS classrooms are noisy and active places that encourage exploration, autonomy, persistence, risk taking, and creativity.

Expanding the Effort

Bringing this kind of computer science instruction into the schools is an ambitious reform that must operate on many fronts simultaneously. Courses must exist in the schools; teachers must be identified and given professional development and support; belief systems about who can do computer science must be continually reexamined; and, state, local, and district policies must support making computer science part of the academic core.

In Los Angeles, one key to getting ECS into schools was the successful petition to the University of California Office of the President for the course to count toward college admission. At the district level, Todd Ullah, former LAUSD Director of Science, helped us design the school implementation procedures.

Principals are asked to identify teachers to attend the professional development at no cost to the school, and the principals then commit to placing ECS on their schools' master schedule. These principals are also asked to identify a diverse range of teachers—especially those most open to inquiry practices and familiar with or eager to learn about computer science. We prioritize underserved schools and those with high numbers of low-income black and Latino students, and counselors help ensure gender equity in the classes. District leadership, local superintendents, and principals' organizations receive ongoing updates about the status of the course.

One introductory high school course is not enough to engage a broader segment of the student population in computer science. The National Science Foundation (NSF) has been supporting the creation of a high school pathway including both ECS and a newly designed advanced placement course called Computer Science Principles (www.csprinciples.org). Like ECS, the new CS Principles course, designed through a collaboration between the NSF and the College Board, focuses on the big ideas of computer science and computational practices. The NSF has also been the launching pad for a CS10K campaign to get 10,000 more computer science teachers into schools (http://cs10kcommunity.org). The Computer Science Teachers Association (http://csta.acm.org); the Association for Computing Machinery (www.acm.org); and others have been active in these and related efforts.

Building on this foundation, the recently formed nonprofit organization Code.org is designing elementary coding courses and middle school curriculum to be integrated into math and science classes, as well as supporting ECS and AP Computer Science Principles at the high school

Video Bonus

See ECS students in action in a video from the National Science Foundation at www.nsf.gov/news/special_reports/science_nation/intotheloop.jsp

level. Code.org has been effective in building momentum around computer science education throughout the United States. The organization has launched a national social media campaign about the importance of computer science for all students, a campaign that has impacted state legislation and school district offerings around the country. (For details, see www.code.org.)

State, local, and district policies have been drafted to make computer science a core class that counts as an academic credit toward graduation. Politicians, including state representatives and the U.S. president and vice president, have been speaking on this issue. President Obama recorded a video message during Computer Science Education Week 2013 encouraging all students to study computer science (www.whitehouse.gov/blog/2013/12/09/don-t-just-play-your-phone-program-it); and Vice President Biden spoke passionately about the need for computer science education at the 2014 National Governors Association meeting.

An Equity Imperative

Computer science knowledge cannot be reserved for elite students who have prior experience, interest, and parental resources. Computer science education must be provided for all students. To broaden participation in computer science programs in schools, we must make it our mission to build talent among all our students, not just to identify talent in a few. As the issue of computer science education gains momentum and attention, it is crucial that a commitment to computer science for all stay in the forefront of these efforts. EL

References

Barton, A. C., & Tan, E. (2010). We be burnin'! Agency, identity, and science learning. *The Journal of the Learning Sciences, 19*(2), 187–229.

California Department of Education. (2012). *California basic educational data system*. Sacramento, CA: Author. Retrieved from www.cde.ca.gov/ds/dc/cb

College Board. (2013). *AP program participation and performance data 2013*. New York: Author. Retrieved from http://research.collegeboard.org/programs/ap/data/participation/2013

Dweck, C. (2007). *Mindset: The new psychology of success*. New York: Ballantine.

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* (pp. 493–498). New York: Association for Computing Machinery.

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.

Jane Margolis (margolis@gseis.ucla.edu) is a senior researcher at the UCLA Graduate School of Education and Information Studies. She is the lead author of *Unlocking the Clubhouse: Women in Computing* (MIT, 2002) and *Stuck in the Shallow End: Education, Race, and Computing* (MIT, 2008). **Joanna Goode** (goodej@gmail.com) is an associate professor in the College of Education at the University of Oregon and a coauthor of *Stuck in the Shallow End*. **Jean J. Ryoo** (jjryoo@gmail.com) is project director of the California Tinkering Afterschool Network and post-doctoral researcher at the San Francisco Exploratorium.