Computer Science for Everyone?

Getting girls to embrace a new basic skill

BY LAURA PAPPANO

As parents, technology advocates and other organizations call for increasing the amount of computer science and coding taught in school, a glaring problem is obvious to teachers across the country: few young women seem interested.

Despite taking math and science courses at nearly the same rate as males, the proportion of female computer science majors in college actually dropped from 39 percent in 1985 to 14 percent today, according to figures from the National Science Foundation and the Computing Research Association.

In high school, just 18 percent of AP Computer Science exam takers in 2013 were female, as compared to 48 percent of AP Calculus AB test takers. Non-Asian minorities are similarly underrepresented in AP Computer Science. In three states, according to analysis by Barbara Ericson, director of Computing Outreach at Georgia Tech, not a single girl took the AP Computer Science exam.

At the same time, Department of Labor figures show the United States producing only enough computer science graduates to fill 61 percent of 1.4 million new computer jobs expected to be added between 2008 and 2018. Still, a 2008 survey of 1,406 college-bound teens by the Association for Computing Machinery and Boston's WGBH public television station found that while 74 percent of college-bound males—regardless of race or ethnicity—thought computer science was a "very good" or "good" major or career, only 32 percent of females did.

The glaring gender gap has experts looking at schools and wondering, Can teaching computer science more broadly and in earlier grades encourage more girls to pursue it? Technology advocates say that making computer science a mainstream academic offering—even a requirement—would encourage and equip more girls to enter the field. Other solutions include changing the image of computing to appeal to girls, actively recruiting female students, and exposing girls to computer science as early as preschool.

Selling Computer Science

The nerdy vibe associated with computer science means that high schools struggle to get *any* students to enroll. The field "has a huge perception problem," says Ainissa Ramirez, who says tech groups like Black Girls Code can counter a prevailing stereotype of the geeky white or Asian male computer wiz.

Yet, computer science skills—from creating and using algorithms to writing software for sorting data or designing a website or app—are essential to STEM fields, says Ramirez, a former Yale engineering professor, author, and science education advocate (see "Computer Science Curriculum," p. x). "More people need to learn how to code than need calculus," because knowing how to harness computing to tackle problems regardless of the field is critical, she says, adding, "You need an introduction."

The College Board, recognizing this need for foundational knowledge for all students, is piloting the AP Computer Science Principles course, with the exam to debut in 2016–2017.

There is no count of how many high schools offer courses. (The Computer Science Teachers Association is conducting such a survey now, with results expected in September.) But in May 2013, fewer than 30,000 students from 3,249 schools took the AP Computer Science exam. In contrast, more than nine times more—some 283,000 from 13,000 schools—took the AP Calculus AB exam.

Can teaching computer science more broadly and in earlier grades encourage more girls to pursue it?

While teachers in most subjects merely type up course descriptions, many computer science teachers realize they need marketing campaigns, video pitches, and personal invitations to attract students, and young women in particular.

Some, like Seth Reichelson, a computer science teacher at Lake Brantley High School in Altamont Springs, Fla., do even more to remake the image of "comp sci." Since he arrived in 2010, Reichelson has driven enrollments in AP Computer Science from zero to 240 by making computer science cool to young men and women alike.

He bought a popcorn machine—the aroma is a draw—and randomly announces availability of low-cost, coveted computer science T-shirts and sweatpants. Also, when Reichelson, an official puppy "foster" parent, holds hours-long challenges like IBM's Master the Mainframe, he brings puppies for students to play with during breaks. And during each of the past two years he has named an AP Computer Science Homecoming King and Queen, who ride in stylish cars (a Bentley convertible in 2013) during the homecoming parade.

In addition to vanquishing the geek vibe, Reichelson also addresses two additional barriers he sees to enrollment: students' uncertainty about what computer science is and the fear that they'll do poorly. He tackles the first issue by finding ways to give students a glimpse at computing in diverse settings. Recently, he partnered with an early childhood development teacher so that students in that class, who also help at the school daycare, could teach coding to preschoolers using Hopscotch, a fun iPad app. While girls enrolled in early childhood education classes typically wouldn't consider taking computer science, Reichelson says that offering this soft introduction has demystified the class and yielded enrollments. Now about 40 percent of the school's computer science students are female.

To make the class less academically risky, Reichelson embraces mastery learning in his AP class, letting students re-do work until they succeed. "There are kids who can write a program for a printer driver in ninth grade," he says, adding that those with little experience "don't want to compete against that kid and ruin their GPA."

More Than Gaming

Another misconception teachers battle is that computer science is all about gaming, which can be a deterrent for female students. Carolyn Mulhern, a computer science major who just finished her freshman year at Penn State, never planned to take computer science while at Deerfield High School in Illinois. "It seemed like a lot of the kids in the class were gamers, and I hate video games," she recalls. She was also scared. "It's material you have never had before."

Her teacher, Steve Svetlik, chair of the mathematics and computer science department, made a personal pitch, and she enrolled, soon discovering that her worries were unfounded. "I don't know why I thought I couldn't do this," says Mulhern, who now works as an online tutor for a website that teaches people how to code.

Initially, when Svetlik found his efforts to enroll students in computer science (offering fun coding demonstrations at lunch) yielded few females, he came to a common conclusion. "I assumed that this was just not something that girls would be interested in," he says. Then he started a girls coding club and began making personal appeals. Enrollments for Introduction to Computer Programming and AP Computer Science have risen to 169 from just 20 a few years ago and from one or two girls to 31 percent and 18 percent, respectively, for the fall.

The personal pitch works, agrees Barbara Ericson of Georgia Tech, who last year e-mailed AP Computer Science teachers across Georgia with a reminder that "it's time to recruit students for the AP CS A class for next year" and offered tips. Among them: "Send personal invitations to those with strong PSAT scores."

That's what Jill Pala, chair of computer science at Girls Preparatory School in Chattanooga, Tenn., has done since 2010. Pala, whose students made up 41 percent of all females taking the AP Computer Science exam in Tennessee last year, believes that it's also critical to confront the image of computer science as a solo pursuit only for certain "types," like geeks.

"Girls don't want to end up working alone in a cubicle. They are social and they want to be involved with the community," says Pala, who recruits beyond traditional math/science stars to expand students' image of who can succeed in computer science. "It is that whole look, 'I'm in the orchestra, she's in the orchestra and she took that class so I can take that class,'" says Pala. "Doors open because they can see themselves" succeeding in the class.

Starting Early

Whatever the cause, girls begin to suspect that they may not be good in computer science or STEM "around third grade," according to Marina Umaschi Bers, professor of child development and computer science at Tuft University near Boston.

That's one reason Bers, director of the DevTech Research Group, decided to teach computer programming and robotics to children in grades K–2. Bers and her team worked with MIT computer scientists to create ScratchJr, a simpler version of Scratch that lets children "write" programs by dragging icons on a touch screen. They also developed KIWI (Kids Invent With Imagination), which lets students program robots by sequencing color-coded blocks (e.g., orange is "sing"; blue is "spin"). Each block has a bar code, so children use a supermarket-style scanner to program the robot, which is a wooden toy with wheels.

Bers piloted the technology and curriculum in 29 classrooms and then tested children's performance on a challenge-based assessment. Early data show no gender differences in success among 272 students using ScratchJr and 245 using KIWI. "Girls and boys performed equally in grades K, 1, and 2," she says.

Bers is not ready to draw conclusions, but she does point out that tasks in the curriculum included activities familiar to girls and boys, tasks involving stories and music. For example, children programmed a robot to perform a dance to music (they could also add a "costume"). "When you think of technology as a way of expression," observes Bers, "that sets the tone very differently."

A 2011 study published in *Computers and Education* showed that even the design of a classroom, whether it reflected computer science stereotypes or not, significantly influenced whether female students decided to enroll in a virtual college computer science course. Researchers saw that varying details,

For Further Information

S. Cheryan, A. N. Meltzoff, and S. Kim. "Classrooms Matter: The Design of Virtual Classrooms Influences Gender Disparities in Computer Science Classes." *Computers & Education* 57, no. 2 (2011): 1825–1835. Available online at **www.usfq.edu**

Computer Science Teachers Association: **http://csta.acm.org**

S. Grover and R. Pea. "Computational Thinking and K-12: A Review of the State of the Field." *Educational Researcher* 42, no. 1 (2013): 38–43.

J. Margolis et al. "Beyond Access: Broadening Participation in High School Computer Science." *ACM Inroads* 3, no. 2 (2012): 72–78.

Ready for Robotics website: **http://ase.tufts.edu/DevTech/ReadyForRobotics/index.asp**

ScratchJr: **www.scratchjr.org**

A. St. John and A. G. Ramirez. *Newton's Football: The Science Behind America's Game*. New York: Ballantine Books, 2013.

such as what hung on the classroom walls—from science fiction items and video game images to nature posters—altered females' attitudes toward the course and feelings about whether they could "belong" and succeed.

Researchers found that including nonstereotypical items increased women's interest in enrolling to the level of their male peers. "Feeling a lack of belonging—a factor that is distinct from abilities or talent—can thus be communicated by even brief exposure to a virtual environment and can have consequences for educational choices, interest, and expectations," they conclude.

For Ria Galanos, a computer science teacher at Thomas Jefferson High School for Science and Technology in Alexandria, Va., one way to raise student interest in computer science is to show how it fits with their passions. When she taught at Centennial High in Roswell, Ga., from 2003 to 2012, Galanos grew computer science enrollments from 15 to 100—*and* got most of the cheerleading squad (she was the coach) to sign up. As cheerleaders, she says, they already had applicable skills. "One thing about cheerleading that may not be obvious," she says, "is we are about routines, procedures, getting things right. We are about the order of things."

At Thomas Jefferson, Galanos has started a girls coding club but says her focus is on helping students see computer science as a useful tool, whether they want to take on water supply issues in developing countries or pursue medicine or banking. "Any strategy I try to use today is not really just about girls," she says. "It's really about showing [students] what the field is about. At the end of the day, it has nothing to do with coding. It is about solving problems." ■

Laura Pappano is a freelance journalist and regular contributor to the Harvard Education Letter.

Quest for Deeper Learning

continued from page 8

the question into categories (e.g., economy, society, politics) for each period. Only then are they asked to discuss the truth of the initial premise.

The precision of this method produces real results for students who stick with it: 100 percent of Downtown Collegiate's students—almost all of whom qualify for free or reduced-price lunch—pass state exams in reading and math, and many go on to attend four-year colleges. The school has thus created a *floor*—a requirement that each and every student will demonstrate proficiency with respect to each and every piece of content the school has identified as important. This floor is an important potential entry point into deeper learning; students are consistently learning the fundamentals of how to analyze texts, how to balance chemistry equations, and so forth.

However, in the absence of opportunities for students to extend what they've learned, there are also significant limitations to this model. After all, the world that Downtown Collegiate's students will enter after graduation is not one in which their tasks will be carefully prespecified. In civic contexts, they will be asked to choose among competing interpretations or to produce their own. In personal contexts, they will face complicated ethical questions. In professional contexts (at least those offering a middle-class wage), they will need to identify the salient features of complex environments and to make arguments for what matters and why. In none of these domains will there be a constant gauge of mastery. And in all of them there will be false starts and missteps—an inevitable part of working under conditions of uncertainty.

Seeking Ceilings

It is these realities that inform the work of a number of other high schools that we studied, schools that are deliberately seeking to align their instructional models with the complexities of modern life. The trade-offs that inhere in such efforts are most apparent in project-based instruction, which shares an ethos with the worlds of invention and design. At schools like Technology High in California, assignments frequently mirror the world described above: tasks are lengthy rather than short and open-ended rather than closed. The socioeconomically diverse students at Technology High have developed a field guide to a local watershed, an illustrated economics textbook (said to have been praised by President Clinton), and "paranoid style documentaries" that link the study of McCarthyism to contemporary social issues. In the course of completing such projects, students learn not only the particular content but also the process of engaging in open-ended work. Taking a cue from Silicon Valley, the school intentionally normalizes failure. Much as industrial designers talk about "failing faster" in order to learn what works, the school expects that over the course of many projects students will sometimes fall short, learning meaningful lessons about the difficulty of producing artifacts of real value.

The strengths and drawbacks of this paradigm are in many ways the inverse of those at schools like Downtown Collegiate. Technology High's students possess a joyfulness about their learning that is not present in most American high schools. At the same time, while the sky is the limit when it comes to excellent work, there is no consistent floor; there is no assurance that all students meet a minimum standard. While the school has pockets of more traditional instruction, much of the learning is "part to whole"; students learn what they need in order to complete specific projects, as opposed to learning a corpus of disciplinary knowledge. This choice is deliberate: In the work world, many things are problem-centered, cutting across fields, and Technology High is similarly