Transforming Computer Science Education in High Schools

Jan Cuny, *National Science Foundation*

The CS 10K Project is an ambitious and bold endeavor that represents our best opportunity to transform computing education beginning where we must begin—at the high school level.

Although high school students routinely use technology—social networking, smartphones, searches for information—their interest in courses that could lead to careers in technology is declining.

The 2009 NAEP High School Transcript Study shows that the percentage of US students taking science, technology, engineering, and mathematics courses has increased over the past 20 years for all STEM disciplines except computer science, where participation dropped from 25 percent to 19 percent (http://nces.ed.gov/nationsreportcard/pdf/studies/2011462.pdf). In a 2009 Computer Science Teachers Association survey, high school teachers reported that they were teaching 8 percent fewer CS advanced placement (AP) courses than just two years earlier (http://csta.acm.org/Research/sub/Projects/ResearchFiles/CSTASurvey09CSResults_DCarter.pdf).

These trends are surprising because projected job growth in IT is very strong—much higher than in all other STEM fields combined—and computational competencies are in high demand in many careers.

There's a renewed focus on K-12 STEM education at the national, state, and local levels, but computer science, for the most part, has been left out. How do we change this?

WHAT'S HAPPENING IN US HIGH SCHOOLS?

The term "computer science" has a specific meaning to university faculty and researchers, but I use it here more generally, as it is often used in the high school context, to include the full range of computing and IT-related disciplines. This generation of students can't be just technology consumers. They must also be technology creators, people who can adapt technology to their own ends and who can express themselves computationally. Thus, all students, whether they're future software engineers and IT innovators or biologists, chemists, engineers, journalists, historians, or artists, will need to know foundational computing concepts.

Currently, rigorous computer science is taught in too few schools by too few certified teachers, and underrepresented groups aren't participating. Consider the following:

- In 41 states, computer science doesn't count as a core course for high school students—that is, because it isn't classified as mathematics or science, it doesn't fulfill graduation requirements (http://csta.acm.org).
- Although some schools do offer CS as a college preparatory elective, most offer it only as a career and technical education (CTE) option.
- High school computer science courses often cover only literacy—keyboarding and the use of software packages.
- Only 14 states have adopted significant standards for computer science education in secondary schools.
- In many school districts, the only rigorous computing course that college-bound students can take is AP CS, but it's offered in fewer than 10 percent of US high schools.
- In 2010, only 14,517 high school seniors took the AP CS test, while 194,784 took the AP calculus AB test, 134,871 took the AP biology test, and 109,609 took the statistics test (www.collegeboard.org).
- CS had the worst gender balance of any of the AP tests. In 2010, only 19 percent of the CS test takers were women, while the percentages for calculus, biology, and statistics were 49 percent, 58 percent, and 51 percent, respectively.

0018-9162/11/$26.00 © 2011 IEEE

During their high school years, students develop problem-solving skills and make initial postsecondary educational choices. Engagement programs to pique and sustain interest in computing in the early grades will be ineffective if those students don't have opportunities to study computing again during their four years of high school. Likewise, enhancing college computing programs will have little impact unless more students are motivated and prepared to enroll in them.

WHAT SHOULD WE DO?

The CISE Directorate of the National Science Foundation (NSF) intends to serve as a catalyst for an ambitious national effort to strengthen high school computing education. The CS 10K Project will support the development of effective new high school curricula and efforts to prepare 10,000 teachers to teach that material in 10,000 high schools by 2015.

The NSF will support research into how students learn computing concepts, and it will assist in the development and evaluation of a range of curricula, course materials, and approaches to teacher preparation and support.

The new curricula will include introductory (or pre-AP) courses as well as a proposed, entirely new AP CS course that the College Board is developing. The curricula will be rigorous and engaging; in addition to imparting computational thinking skills, they will expose students to the breadth of application and the "magic" of computing. The introductory courses will be designed for both college preparatory and career and technical education students.

The proposed new AP CS course—called CS Principles—will be designed both for students with interests in CS and for those with more general interests in science, engineering, and the humanities (http://csprinciples.org). The College Board will also continue to offer the existing AP CS test.

WHY SO MUCH FOCUS ON AP?

Education in the US is decentralized, with curriculum determined at the state and local school district levels. That makes it difficult to effect the widespread change that's needed.

AP courses are attractive to schools because they increase their academic profile, to students because of their potential to strengthen transcripts, to parents because they can reduce college expenses, and to college admissions officers because they are known quantities.

Although it does include programming, the CS Principles course isn't programming-centric.

The AP imprimatur will thus greatly facilitate adoption of the CS Principles course. More importantly, it's our only single point of national leverage, and it carries with it assurance of fidelity of replication.

SIGNS OF PROGRESS

Curriculum development for the new courses is well under way. With NSF funding, a College Board Commission, working with a distinguished advisory committee, has designed the framework for the CS Principles course. Although it does include programming, the course isn't programming-centric. Instead, it focuses on the underlying principles of computation including problem solving, abstraction, algorithms, data and knowledge creation, and programming.

The course covers the limitations of computation and exposes students to both its breadth of application and related issues of society, culture, and ethics. It's intended to be inspiring, to emphasize creativity, and to be relevant to a diverse group of students. The framework has recently been sent to colleges for vetting, and the resulting comments and suggestions will be incorporated into a final version of the framework this summer.

Progress has also been made in developing curricula that implement the framework. During the 2010-2011 academic year, CS Principles courses were piloted at five universities (AP courses are, after all, college-level courses): Metropolitan State College of Denver; the University of California, Berkeley; the University of North Carolina Charlotte; the University of California, San Diego; and the University of Washington. These courses were different, even in the languages they used: Scratch, BYOB Scratch, Alice, and Processing.

During the 2011-2012 academic year, we expect that additional pilots will be run at 20 to 25 universities and 20 to 40 high schools. SIGCSE, the ACM Education Board and Council, and 100 universities and colleges have endorsed the CS Principles course.

The curriculum for the pre-AP courses will be more flexible, allowing teachers to define their class locally. To assist, the NSF will support the development of a curricular framework, instructional materials, and a set of exemplars. One of those exemplars will be the new Exploring Computer Science (ECS) course, also developed with NSF funding. ECS was piloted this year in more than 20 schools in Los Angeles, San Jose, and Oakland. Its complete curriculum is available on the CSTA website (http://csta.acm.org).

WHAT ARE THE CHALLENGES?

Curriculum isn't our biggest challenge. The biggest challenge will be developing effective teacher preparation and support, and scaling it to reach 10,000 teachers. Few schools today have teachers with any formal

CS training. The computing community must launch an unprecedented effort to prepare teachers, working with in-service as well as pre-service teachers, and in both traditional and alternative certification programs. We'll need to pair face-to-face training with extensive, state-of-the-art online support that includes curricula, instructional materials, assessments, and social networking.

The NSF intends to catalyze this effort through its Computing Education for the 21st Century (CE21) program. To scale the effort to 10,000 schools, though, we'll need more than just NSF support. Success will require establishing a public/private partnership to provide the significant additional funding needed to reach 10,000 teachers. Advocacy groups will also need to address state and local regulations on teacher credentialing, curriculum standards, and the placement of computing within graduation requirements.

The magnitude of this project is daunting. To be successful, we'll need to enlist the entire computing community in the effort—K-12 teachers, university faculty, and undergraduate and graduate students in service learning and outreach programs, as well as professionals serving as citizen scientists.

The CS 10K Project is ambitious and bold, but the situation is dire and not getting any better. CS 10K is our best shot at transforming computing education. Its success will be critical to our nation's ability to innovate and compete in the global 21st-century economy.

Jan Cuny is a CISE program officer, National Science Foundation. Contact her at jcuny@nsf.gov.

Editor: Ann E.K. Sobel, Department of Computer Science and Software Engineering, Miami University; sobelae@muohio.edu