

BLOG@CACM

The Danger of Requiring Computer Science in K-12 Schools

By Mark Guzdial
April 15, 2014
[Comments \(5\)](#)

VIEW AS:   SHARE:                    



Code.org has changed how people think about computer science in schools in the United States. Their [successful video](#) and [Hour of Code event in December](#) has attracted enormous attention. There are more calls for computer science in schools, and [more states are making computer science count](#) towards high school graduation (including some odd efforts to [make computer science count for foreign language credit](#)).

Some are now calling for computer science to be a *required* subject for U.S. school children. [Lawmakers in California are considering Bill AB 1530](#) which would add CS to the required course of study for grades 1 to 6 (roughly, six to nine years old). [CIO.com quoted Ashley Gavin](#), the curriculum director at *Girls Who Code*, insisting that computer science be mandatory in schools. "You make it an option, the girl is not going to take it. You have to make it mandatory and start it at a young age."

We are not ready to make computer science a requirement for all children in the United States — even if it's where we might want to be one day. We don't know how to do it, and if we simply made it a mandate today, we would not achieve our goals.

I have written before about [how far behind other STEM subjects we are in computer science](#). In the U.S., maybe one high school in ten has a computer science teacher. Far fewer schools serving grades 1-6 have CS teachers. If we mandated computer science tomorrow, where would we get the teachers? Who would to teach CS to so many teachers?

And what would they teach? I recently spoke with [Roy Pea at Stanford](#) about the California legislation. Roy was [one of the first to study how children learned to program](#). He said that when schools first started creating programming classes in the 1980s, teachers often didn't know much about programming themselves. So, they tested students on what they could test. One of Roy's examples was, "What are the dimensions of a 5.25 inch floppy disk?"

Here in Georgia, we were one of the first states to use the [CSTA Model K-12 Curriculum](#) to design a computer science set of courses ("pathway") in high school. The initial course was "[Computing in the Modern World](#)." But soon after adoption (2007), the professional development budget was cut dramatically. Too few teachers learned to teach the new courses. As the curriculum were revised, the learning objectives were lowered. Most of the CS content was removed. The new (2013) initial course is "Introduction to Information Technology," and learning objectives now include the skills necessary to run a customer support call center ("[Determine the best method to maintain a customer list and communication platform](#).")

While we can argue that [computing is important for everyone](#), requiring computer science in K-12 schools really means *everyone*. Tony Dillon in South Carolina's Department of Education worries about the lower end students. South Carolina already has a requirement that every student must take computer science, but as Roy Pea would predict, the requirement can currently be met with courses in CAD or Photoshop. Tony is worried about the impact on special education students of raising the CS standard. For these students, a high school degree is a challenge, and if they are successful, the degree helps them get a job. Do we know how to teach CS to students with learning or development disabilities? Can we confidently state that, without CS, those students should *not* earn a high school degree?

I doubt that Ashley Gavin is right. Girls might take CS if it's available. Studies like [Stuck in the Shallow End](#) show us how female students and members of under-represented minority lack access to CS classes. Our first step is to provide access. If computer science counts towards high school graduation, then schools have a reason to offer it. We can [improve \(in most states, "create"\) CS teacher certification](#) to offer schools a way to identify well-prepared teachers. Will a girl choose to take a high-quality CS class as an option for meeting a math or science requirement? I think so. [A single all-girls school in Tennessee sent so many girls to the AP CS](#), they made Tennessee the top state in the US for female AP CS exam takers. When 90% of schools don't offer CS *at all*, we can't know how much will change if we provide access.

The argument for [requiring computer science of all higher-education students](#) is much stronger. Every campus with a Computing department has faculty at-hand who know fundamental CS — no professional development needed. Many (if not most) of [our undergraduates will likely need knowledge of computing including some programming](#) at some point in their future careers. [There is likely a productivity cost of not having that knowledge](#), even for today's college-educated workers. If we can't make the case on our campuses, when the argument is so much stronger, why would we think that it's better or easier to push CS on all the K-12 schools in the U.S.?

Comments

Justin Megawarne

April 20, 2014 06:29

I dread the introduction of programming into the curriculum in the UK. One headteacher has already informed me that her staff have received their one-day training in Scratch, and are therefore now qualified to teach the subject. At best, they'll be afraid; at worst, they'll be deluded.

It will all end up like maths class: enjoyed by a lucky few and hated by everyone else. The subject has been distorted, with poor educators hammering pupils for getting the wrong answer. What can you expect when their own performance targets are riding on exam results?

The answer lies with adults in the real world. Adults have to show a personal interest in learning programming and apply it for fun and profit. If they dont, kids wont see what the point is. It'll be another artifice of the school system to them, like high-school trigonometry or mediaeval England. But this is an inconvenient answer, because it means we have to take responsibility for failing kids ourselves, whereas now we have teachers to be our scapegoats.

As a society, we see the education system as a magical way to create better adults than we were. We are like a competitive father who forces his son to play football, because he missed his own chance to join the youth team as a teenager. But does anyone care whether the children are interested? Why should they learn something that we dont seem to value ourselves? I see society-at-large reading and doing arithmetic, but I dont see them programming. For that matter, I dont the typical adult enjoying a discussion of the cultural consequences of Mughal India, worrying about the problem of favelas in Brazil, piqued by how photosynthesis converts light into energy for a living organism, or any of the other actually incredibly interesting things that I was forced to learn in school at exam-point. The net effect is that the adults dont appear to value these things, but they tell the children to cram it in, lest they end up destitute.

So if theres anywhere that we should introduce a computer science curriculum, its in our own lives. And if we dont want to do that, well, then, we should shut up about getting kids to learn it.

Douglas Blank

April 22, 2014 11:56

I think Ashley is correct. She said that if "you make it an option, the girl is not going to take it." That is, many girls will opt out of a computer science course if you give them the chance. We don't even offer an introductory CS course at Bryn Mawr College for non-majors for this very reason. In fact, if you make CS optional and more widely available, chances are that you'd see an even larger disparity between boys and girls in CS than we see now. Unless you want to see a growing disparity when courses become more widely available, the only alternative is to "make it mandatory".

Many women don't discover that they enjoy CS until college. That is too late for many. It makes sense to try to fix this to "start it at a young age."

Yes, we don't have enough teachers. Yes, we don't know how to teach it (or Math, Physics, or Latin either, many would argue). Yes, many students hate and don't appreciate what is already required. But if you restructure what is already taught in K-12, and revise how it is taught, you might find that CS principles fit nicely in many areas.

Mark Guzdial

April 25, 2014 02:44

Doug, I think Jill Pala would disagree. She's the AP CS teacher in Tennessee whose 30 female AP CS exam-takers in 2013 gave Tennessee the distinction of having the most females in AP CS in the US (<http://www.timesfreepress.com/news/2014/feb/24/gps-teacher-responsible-states-top-rank-female-stu/>). They weren't required to take CS. I'm sorry that Bryn Mawr doesn't offer intro CS to non-majors. You might reconsider. Stanford's course isn't required for non-majors, but 90% of all Stanford undergraduates take it (<http://mashable.com/2012/07/01/stanford-top-major-computer-science/>).

I totally agree that we need to make CS accessible at a younger age. Making it mandatory will force the majority of schools to invent something that they'll call "computer science," but it will look like Photoshop or CAD software instruction because that's all that they can do without a CS teacher. Let's work first at making it accessible, before we try to require it. We don't even mandate a particular mathematics level for everyone nor any specific science (e.g., Physics or Chemistry). How can we argue that computer science is more important than all the rest and demands a mandate that the others do not deserve?

Doug Lewin

April 22, 2015 12:43

Based on Marks comments, which is right on for the most part, looks like the chicken or egg dilemma, which came first. Why not both at the same time. Have states require Computer Science at some date in the future in high schools and have milestones they need to reach along the way. This forces school districts to put computer science on the radar map. In my district teachers tend to be the ones driving new courses. Thus if current teachers cannot teach computer science as Mark states, what makes us think that districts will have interest in creating computer science classes. At the same time require every college who credentials teachers to establish a computer science credential. In California the best you can do is get a supplemental certificate from the state. Sometimes you have to push while you are pulling.

Mark Guzdial

April 22, 2015 10:23

Doug, I agree — that's probably the fastest way to get computing into schools: Make it a requirement, and for all districts who can't meet the requirement at first, require a plan to get there. That plan may have to be multi-year. At least here in the US, we simply don't have enough CS teachers to provide every school with one today. It will take time and funding to get there.

Justin, you're absolutely right that the best way to get kids interested in computing is to immerse them in a culture, including many adults, who program. Seymour Papert said this in his 1981 "Mindstorms," and he praised the Samba School culture in Brazil as a model of multi-generational, immersive learning. He saw Logo as a way to create a "Mathland" where students could be immersed in mathematics and create a culture that used and appreciated mathematics.

Just because kids don't like something at first isn't a reason not to promote it, for the value of being literate and being a full participating citizen in a democracy. When Alan Perlis suggested teaching computing to everyone on campus in 1961, Peter Elias (chair of electrical engineering at MIT at the time) pushed back. Programming was an awful bore and tedious. Wouldn't it be better to just wait until programming became easier? J.C.R. Licklider (who later went to DARPA and funded the development of the Internet) responded: "Peter, I think the first apes who tried to talk with one another decided that learning language was a dreadful boreBut some people write poetry in the language we speak."□

Displaying **all 5** comments

SIGN IN for Full Access

[» Forgot Password?](#)

[» Create an ACM Web Account](#)

SIGN IN

- MORE NEWS & OPINIONS
- Computer Model May Help to More Accurately Predict Volcano Eruptions**
BBC Science Focus
- Why Robot Umpires Are Inevitable in Baseball's Future**
Tech.pinions
- What Help Should We Provide to Students Learning to Program?**
Mark Guzdial

- ACM RESOURCES
- Recognizing and Managing Anger (Second Edition)** 
Courses