

Should Computer Science Classes Be More Practical?

smartbear.com/blog/math-nerds-vs-code-monkeys-should-computer-science

Math Nerds vs. Code Monkeys: Should Computer Science Classes Be More



- “Many employers don't care what courses are on your transcript or even what your GPA is. Instead they want to know what you can do, and especially what have you built.”

Good advice, says software engineer and author Gayle Laakmann McDowell, whose books offer tips for people looking for jobs in the tech industry. “There are library books out there; pick up some to learn the basics. Then build something on your own.” McDowell recommends sites like www.codecademy.com for students learning new programs.

Which begs the question: *Why aren't computer science students learning these skills in college? Should computer science programs teach more practical skills?*

“The ideal program has theory but is backed in programming and practical applications. There are programs that are too theoretical and others that are too practical,” McDowell says. If a school is too oriented on the practical and emphasizes programming too much, students aren't challenged on solving complex problems.

Another potential complication: “They’ll get tied too much to today’s hot language, and in 10 years, it’ll be a dead language.”

Large companies offer training to new hires, but usually newbies start writing code within two weeks, if not on Day 2.

“At Apple, I wrote code on the third day there, never having used that program before,” McDowell says. “You need to pick up coding languages easily. It’s fairly easy to learn a language if you know a couple of others.”

Computer science grads who throw themselves into the soup of real software engineering will most likely pick up the skills they need after a few weeks of nervousness.

“Coders Should Learn By Coding”

But not everyone is happy with this state of affairs. The demand for people who can code is urgent and growing every day. Is the four-year degree in computer science that most employers insist on really the best way to satisfy it?

No, says Jeffrey McManus, creator of an online course called www.CodeLesson.com. McManus wrote a post for his personal blog entitled [More Universities Should Shut Down Their Computer Science Programs](#).

“Computer science focuses on theory. It comes from math departments,” McManus said. Most students, on the other hand, want to learn applied software engineering, not computer science. Like McDowell, McManus says that the ideal program would offer a balance between the theoretical and the practical. But he says most computer science courses aren’t very good at teaching the up-to-date skills necessary to the field because the professors received their degrees years earlier, when tools and tactics were different. Software skills are best taught online, he argues.

“Coders should learn by coding, not by sitting in classrooms listening to lectures,” McManus says. And developers can’t expect to learn programming skills from their employers, who are increasingly less willing to invest in employee training.

Some computer science students agree. Sheldon Kreger, who studied computer science but dropped out before graduating, is now a successful web developer at www.OpenSourcery.com. He learned coding on his own and from other students, and continues to learn from colleagues.

“It’s widely understood that computer science doesn’t necessarily make you a good programmer,” Kreger says. “It gives you theoretical knowledge, but because there are so many topics, you skim through everything quickly, and a lot of people come out not being very good programmers.”

Some students he met *did* know programming, but they had learned it outside of school. Kreger says he learned a lot from working in groups with them.

The professors, on the other hand, were busy with administrative tasks and didn't help with coding. And the classes? Here's Kreger's comment on McManus' blog:

“Sure, I learned what a stack is and coded one up by hand, but I can learn what a stack is in about 5 minutes from a book. No client cares if I ever coded up a stack, and if I needed to for some reason, I would pull it out of a pre-existing library. But, learning about vectors, stacks, and linked lists took an *entire semester*, and hundreds of dollars of tuition money. Seriously?”

Don't Fly My Plane

On the other hand, some former students think the theoretical and mathematical parts of their education are extremely valuable.

“If something starts to break and you don't understand what's happening, much less why it's happening, you're in trouble. The people who took computer science know what's happening under the covers and can diagnose when something goes wrong,” says Jaime Torres, whose degree in electrical engineering included solving complex computing problems. He now works for data mining company ICF International doing content management and working with Sharepoint and ASP.net.

Torres concedes he might be able to do his job without the degree, “but I don't think I'd be as effective,” he says. “Code monkeys” who dot the *i*'s and cross the *t*'s are necessary in today's software-driven world, he asserts, “but to have them writing the software that's flying my plane... I don't want them doing that.”

The Billionaire Dropouts

Many students enter the computer science field as a route to career success. But it's hard for them to close their eyes to the impressive roster of successful college dropouts, including billionaires Bill Gates, Paul Allen, Steve Jobs, Michael Dell, Larry Ellison, and Mark Zuckerberg.

And of course, PayPal founder and venture capitalist Peter Thiel, who told *60 Minutes* that everyone *thinks* they need a college degree just like everyone *thought* they had to have a house or two during the housing bubble, but the reality is that higher education is overpriced and often unnecessary. His 20 Under 20 fellowship pays \$100,000 to 20 students every year to drop out of college.

Ed Lazowska, who holds the Bill and Melinda Gates chair in computer science at the University of Washington, says that dropping out and learning on your own can work for some, but it's not suitable for most students. One pragmatic reason is that few companies will hire software engineers without a degree.

But there's something more fundamental, something he calls “thought skills,” acquired from working in teams and doing things, not from a programming book.

“The further out you are from a degree, the less important it is what you majored in. What’s more important is the stuff your mom told you about and you rolled your eyes: critical thinking, analysis, the ability to speak and write. Any top bachelor’s program will teach these. In the end, that’s what makes people successful.”

Few would disagree about the importance of these so-called soft skills. But while people skills and critical thinking require a group setting and life experience, colleges are turning more and more to the online world to teach skills like programming.

The Online Tsunami

More than a dozen colleges and universities, including Princeton, Stanford, the University of Pennsylvania, and Duke, are partnering this year with www.Coursera.com, a new company founded by two Stanford computer scientists that provides online courses, some of which now provide credit. Offerings are expanding, but most of the initial classes were in computer science, math, and engineering. The response has been what one professor called “a tsunami.” Last year, before the expansion, the company taught 43 courses and registered 680,000 students from 190 countries with its original partners (University of Michigan, Princeton, Stanford, and the University of Pennsylvania.)

Coursera is only a year old and is not yet self-sustaining economically, though it hopes to be at some point. Currently, its contracts state that if a revenue stream emerges, it will be shared between it and its university partners. Some colleges have questions about the program and are reluctant to climb onboard. McManus, whose company partners with the University of Victoria in Canada, says some institutions have built-in resistance to outside teachers and courses.

But given the massive interest in the program and the proliferation of sites like McManus’ CodeLesson, Codecademy, and www.Lynda.com, it seems likely that online education, whether it comes from massive-enrollment courses from Coursera or from smaller courses offered by other companies, will increasingly form part of students’ college experience, particularly when it comes to fields like computer science.

If so, practical courses offering hands-on coding experience and exposure to hot new languages could have a place on campus without interfering with basic and theoretical classroom instruction. It seems only appropriate that the debate between the theoretical and the practical in computer science education will eventually be resolved online.

See also:



**Can I Really Roll-out
a Code Review Process
That Doesn't Suck?**

Free on-demand webinar
Learn how to get started with code review,
be successful, and have great reviews!

View Webinar Now 

Like what you've read? Click here to subscribe to this blog!