IDEAS

Should Computer Education Cover More Than Just Coding?

Updated February 26, 2016 · 3:08 PM ET
Published February 15, 2016 · 7:00 AM ET

BYRD PINKERTON

Ammar Al-Kahfah plays with a stuffed "Baymax" toy at the Georgetown Hackathon in Washington, D.C. His team has wired it to move and to collect basic medical information.

LA Johnson/NPR

President Obama wants kids to learn to code. So much so, he's pledged billions of dollars to teach them.

"Now we have to make sure all our kids are equipped for the jobs of the future – which means not just being able to work with computers, but developing the analytical and coding skills to power our innovation economy," he said in his radio address on Jan. 30.

And adults are looking to learn, too. Coding academies, or "boot camps," are cropping up across the country, promising to teach students to code in a few months or even a few weeks.

But computers are not just about coding. There's also a lot of theory — and science — behind technology. And those theoretical concepts form the basis of much of computer science education in colleges and universities.

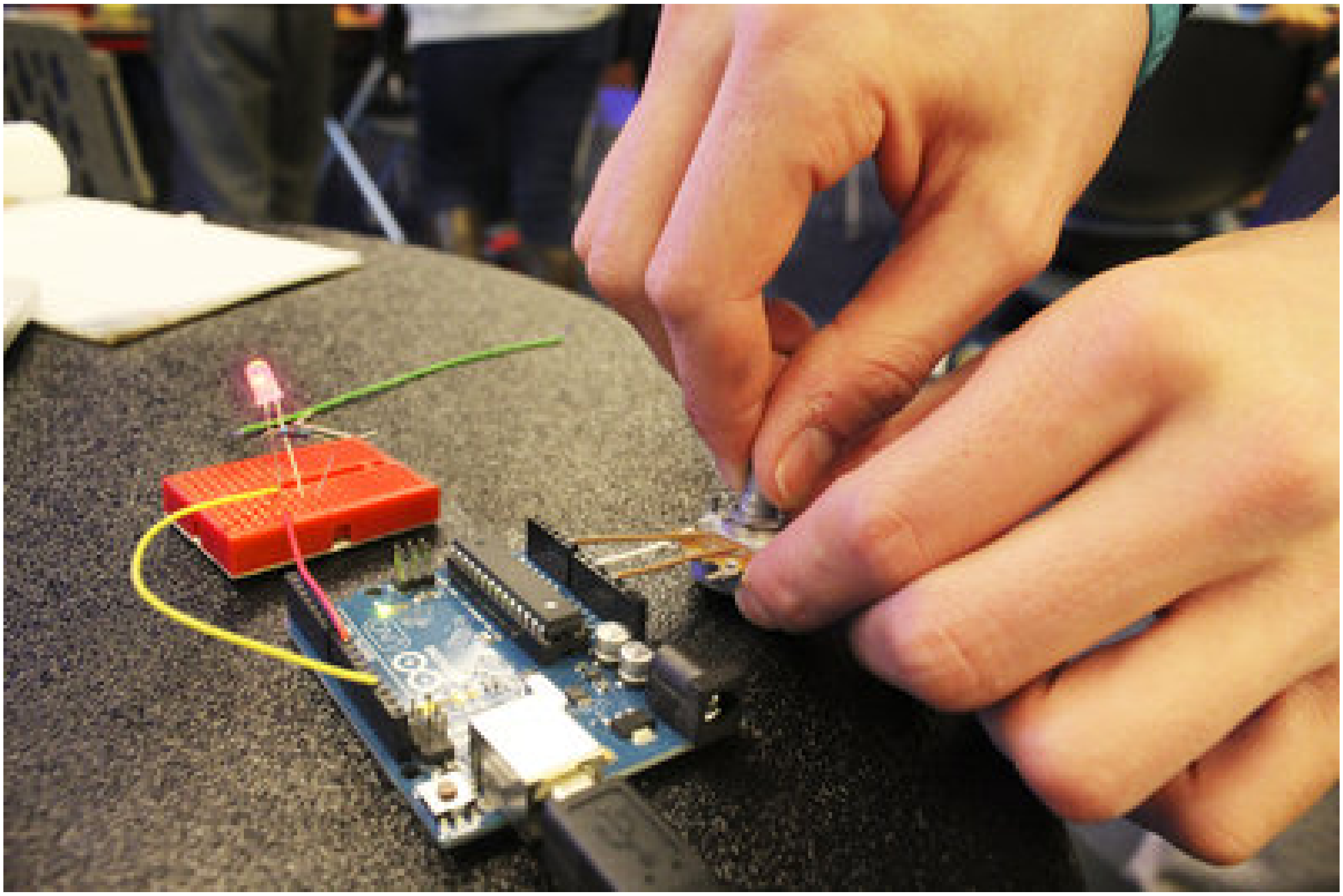
Lisa Singh, an associate professor at Georgetown University, stands behind that theoretical approach.

"We now need to train everybody to understand the basics of computer science," she says, "and I don't equate it to just coding. I equate it to principles of thinking."

There are ways of approaching problems, for example, or of structuring data, that help students program more effectively and more thoughtfully.

If coding boot camps may downplay the theory, though, Singh admits that many university classes are heavy on it.

At Georgetown, classes in cybersecurity or data structures train computer science majors how to use mathematical proofs to solve problems and to understand the limitations of their solutions.



A student works on a light sensor during an Arduino workshop at the Georgetown Hackathon in Washington, D.C.

LA Johnson/NPR

Singh doesn't think that non-majors need to cover all of these concepts. But there's a certain way of approaching computers, something she calls "algorithmic thinking," that she thinks every student should learn. She wants to teach them to break problems down into a series of steps, an approach that requires some theoretical knowledge.

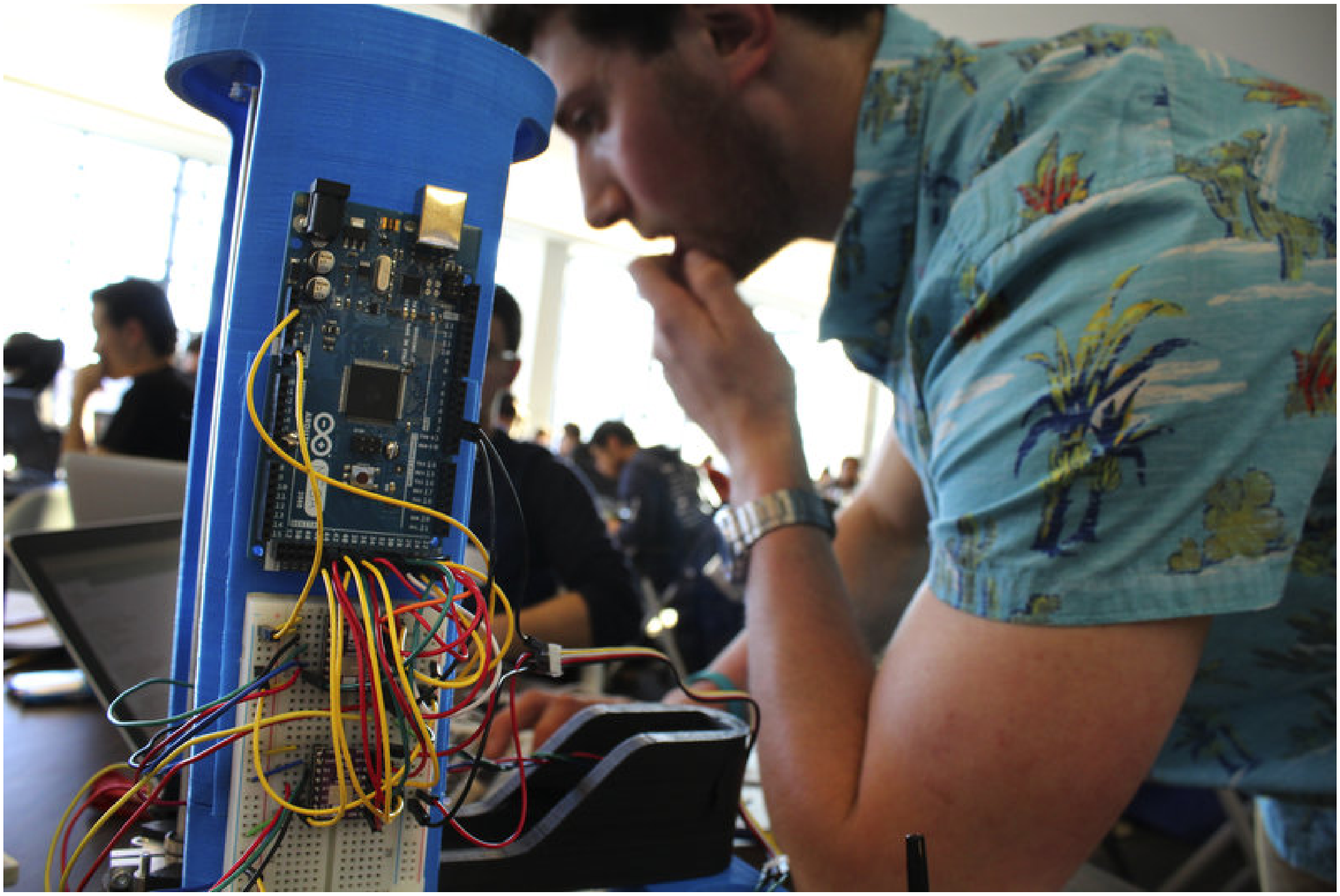
At the end of their education, she thinks students should know, "what the heck is an algorithm and why do I care about an algorithm," she says. "Because if you don't understand that, the fact that you can code something up, it doesn't have the same meaning to you. You're not thinking about that problem the same way."

Not everyone agrees.

Gene Chorba, a student at Georgia State University, works full-time for a Georgia-based startup.

"In the traditional collegiate learning experience," he says, "you sit in a classroom, you're given a book you spend hundreds of dollars on, and most of the information you learn is never used."

He says his company has stopped recruiting at career fairs. Instead Chorba does recruiting at hackathons, where students code applications and gadgets they can show prospective employers.



David Kaplan, a University of Maryland, College Park sophomore, tinkers with a robotic arm programmed to play tic-tac-toe at the Georgetown Hackathon in Washington, D.C.

LA Johnson/NPR

He was at a hackathon last month at Georgetown that drew more than 300 students from all over the country for a 36-hour event.

A sophomore from the University of Maryland, Jenny Mandl, was there, too. She disagrees with Chorba's views.

"There's so many things that you learn in your classes that you wouldn't have otherwise gone to find," she argues. "That's one of the main things that creates creativity."

For their hackathon project, Mandl and three classmates worked on a robot designed to play tic-tac-toe. It's a mix of high-tech, with a mechanical arm made from a 3-D printer, and low-tech — rubber bands and electrical tape.

They're trying to get the bot to read the moves a player makes on a paper game board, and then to draw its own moves in response.

All of this requires programming, which isn't emphasized in the classroom. But Mandl insists that she needed the skills and techniques she learned in her courses, too.

OLDER
A Symphony's Big Challenge: Lift A Tough School Through Music

NEWER
Standards, Grades And Tests Are Wildly Outdated, Argues 'End Of Average'

About
Subscribe
Connect With Us

SHARE