

COMPUTATIONAL THINKING IS BECOMING ONE OF THE THREE RS

By Charles Day



“I’M LOSING MY VOICE TALKING ABOUT MATLAB AND C++. HOW WILL MY CADETS BE TRUSTED IN LEADERSHIP POSITIONS IF THEY DON’T UNDERSTAND HOW TO PROGRAM?”

Thus ran a recent Facebook rant from a friend who teaches at a US military academy. His not-quite-serious exasperation reminded me of a complaint from 20 years ago from another friend, Robin.

Robin still is, and I used to be, an x-ray astronomer. Back then, the programs we ran to analyze our data were on the brink of becoming plug-and-play packages that could handle data from various spacecraft and run on the operating systems of various platforms. Robin worried that the graduate students would become adept at using computers without learning how to program them.

I’m not sure if Robin’s worries came to pass. The students I encounter when I visit labs or go to conferences do seem to know how to program. At the University of Science and Technology of China in Hefei, I met two students who were working on a computational physics project related to atomic collisions. When I asked their adviser how they acquired their computing skills, he replied, “I don’t know. They study computing for fun.”

If you’re in graduate school, you have to produce a thesis, which entails doing new and original research. In a data-intensive field like astronomy, “new and original” will likely involve doing something on a computer that no one else has done before. You’ll probably have to write a program. I had to.

But the cadets my Facebook friend teaches aren’t scientists. They won’t need the ability to write code. On the other hand, they and other young people who aspire to challenging and engaging careers will need what writing your own program entails: the ability to frame a nontrivial problem and solve it by putting together a sequence of logical instructions.

The ability to see a problem and then to solve it is important not just in science and the military. Lawyers need a similar ability when putting together and arguing a case. Authors need it when conceiving of a subject and then putting it into words. Like other kinds of writing, programming requires clarity of thought and expression.

As computers become more pervasive in the classroom, there’s a temptation to favor those teaching methods that can be most readily automated. If teachers succumb to that lure and abandon the practice of setting and marking essays, then perhaps programming, which is intrinsically digital, might remain one of the few vehicles for teaching students how to think. After all, a program’s inputs and outputs are easier to assess than an essay’s arguments.

A commenter on Robin’s Facebook rant wrote, “Computational thinking is becoming one of the three Rs.” In my view, computational thinking, or something like it, has always been, and always should be, essential to education.

Charles Day is *Physics Today’s* online editor. The last time he wrote code was in 1998, when he programmed his HP 42S to calculate mortgage payments.



Selected articles and columns from IEEE Computer Society publications are also available for free at <http://ComputingNow.computer.org>.

This article was featured in

computing **now**

ACCESS | DISCOVER | ENGAGE

For access to more content from the IEEE Computer Society,
see computingnow.computer.org.



IEEE

IEEE  **computer society**

Top articles, podcasts, and more.



computingnow.computer.org