Dystopia

By Isabel Beichl, Editor in Chief

As I noted in my last column, for several of the past 10 years, I’ve helped to organize and manage an undergraduate summer program sponsored by the laboratory where I work. Anyone who has acted as a mentor in such a program will tell you it takes a lot of effort (“Skateboarding in the halls is strictly forbidden!”), but those efforts are rewarded in the joy of working with bright and eager young scientists (“This student writes more code in a week than I do in a month!”).

One side effect of my involvement is that I get a glimpse into the evolution of undergraduate sciences and engineering education. In some areas, things are changing more rapidly than I’d guessed; in other areas, change isn’t nearly fast enough. Both trends are mostly healthy. In this imperfect world, however, there is (as usual) some cause for worry.

On the too-fast side, there are now majors in topics such as gaming and Web design. These topics are, in my humble opinion, well worth learning something about—and might even provide a way to earn a living—but in and of themselves surely don’t constitute an education. On the too-slow side, I was slightly dismayed to discover that some mathematics majors at first-rate universities can graduate without having taken a course in computing.

In their foundational document, “Preliminary Discussion of the Logical Design of an Electronic Computing Instrument,” Arthur W. Burks, Herman H. Goldstine, and John von Neumann offer a treasure trove of information for those interested in the history of digital computers. It contains many of the basic ideas that have driven the development of programming and computing, including the notion of the stored program machine. Although the concept had certainly been invented earlier (by Turing in 1936, Zuse in 1936, Eckert and Mauchly in 1943, and so on), the Burks-Goldstine-von-Neumann report had the greatest influence.

An interesting feature of the report is its discussion of possible uses of computing devices. The usual things related to modeling of physical situations are, of course, mentioned, but there’s also material on non-numeric, combinatorial applications. Such applications are now heavily used by some mathematicians working in fields that were traditionally called “pure,” including algebraic topology, study of groups, and graph theory. Crucially, these applications aren’t based on symbolic computing (which is also widely used), but rather on use of data structures and combinatorial algorithms to test out conjectures and construct examples as an aid to proposing and proving theorems. To do this effectively and productively, researchers must have a fairly deep understanding of data structures and computational complexity’s practical meaning. The investigations can’t be carried out merely by running an existing script on Mathematica, Matlab, R, and so on. Often, they must build things from the ground up. Doing so requires a solid background in computing.

The author Gary Shteyngart, in his beautiful recent novel Super Sad True Love Story, imagines a dystopian near-future in which the world is much more digitally connected than our world today. In Shteyngart’s world, you can get a profile of another person simply by pointing something like an advanced iPod in the direction of the target person’s iPod. Individual credit ratings are common knowledge, posted every time someone walks by a “credit pole.” Almost nobody reads printed matter, but everyone streams stories and texts to friends. People “verbal” one another rather than talking, and one of the most socially elevated jobs is fashion store clerk.

There is, however, one false note in Shteyngart’s amazing construction. There’s no mention made of how the technology was created and none of the novel’s characters exhibit any knowledge of how it works. In short, the main characters are sensitive and intelligent college graduates, but they are not educated.

I thank Francis Sullivan for his interesting comments and insight on this subject.