Vinton G. Cerf

DOI:10.1145/2492007.2492009

Computer Science Education—Revisited

ACM members span a remarkable period in the history of computing. We can point to Kelly Gotlieb, chair or co-chair of the ACM Awards committee for over 20 years, who joined ACM

in 1949 when it was two years old! Next to Kelly, I feel like a newbie, having joined in 1967. And there are the most recent members, who joined us just this year, many as a result of our international initiatives.

We have a wide range of stories of the academic and professional paths that have led us to become practicing members of ACM and our profession. Similarly, our educational experiences are equally varied. Many of our members received degrees in disciplines other than computer science simply because their academic years preceded the creation of computer science departments. Indeed, Purdue University recently celebrated the 50th anniversary of its computer science department—the oldest CS department in the U.S.

At this stage of computing's evolution, it seems appropriate to reflect on what we should learn about computing and how we can or should learn it. I prefer to say "learn" rather than "teach" simply because the important metric is what one can learn about the field. Many of our most productive colleagues joined the computer science or computing field from other disciplines. Physics is often a precursor to a career in computing. Large-scale simulations, computational biology, computational linguistics, and other computational analogues along with "big data" processing occupy a good deal of the attention of today's software and hardware practitioners. Subtle and deep analyses drawing on Bayesian reasoning also oc-

cupy the thoughts of many of our colleagues.

But what about people who have no intention of entering the field? Shouldn't they have some idea of how programming works? How operating systems work? How networks and their layered protocols work?

Among the initiatives that ACM has been pursuing is to make computer science acceptable as a core science along with mathematics, physics, biology, and chemistry. This is especially critical in secondary schools where, with few exceptions, computing classes tend to be optional and not a substitute for any other discipline. In many advanced programs, it is a requirement to have a certain number of credits in science, for example. It is ACM's position that

In many advanced programs, it is a requirement to have a certain number of credits in science. It is ACM's position that computer science should have equal standing.

computer science should have equal standing. Moreover, the curriculum should include some serious exposure, inter alia, to programming, systems, languages, and computer architecture. The idea is not necessarily to turn students into professional computer engineers and scientists, but to expose them to the richness of computer science and to help them appreciate the potential nascent in computers and programmable systems.

A detailed report on this topic can be found at http://csta.acm.org/Advocacy_Outreach/sub/Inroads_Stephenson-Wilson.pdf. I strongly urge you to read this report and then to explore ACM's website page on education: <http://www.acm.org/education>.

Reforming K–12 education to incorporate serious computer science seems vital to producing an informed public that has a deeper appreciation for the power of computing than video games and social networking. There are, no doubt, countless opportunities for computing professionals to engage in this effort, by lending their support and time to the effort to reform K–12 curriculum content and to make visible to young people the excitement of discovering what computing can accomplish. The discipline of writing and debugging software, of creating simulations or interactive applications has the potential to draw many into the profession, or at least to provide even more with a sense of the core role computing is playing and will play in the decades ahead.

As the Internet of Things becomes reality and software appears in every appliance, building, and vehicle, we have a societal interest in promoting understanding of and interest in our discipline.

Vinton G. Cerf, ACM PRESIDENT