Lynn Arthur Steen

Mathematics and Computing Education: A Common Cause

For many years, especially as computing has moved beyond the immediate influence of scientific computation, the mathematics and computer science communities have been traveling in separate orbits, only loosely coupled by common concern for algorithms and discrete processes. In universities and industry now, computer science and mathematics are essentially separate disciplines.

In education, however, mathematics intertwines with computer science—in training teachers, in developing curricula, in attracting students. By some estimates, between 50 and 70 percent of all lower-division college-level computer courses (programming and elementary computer science) are taught by individuals who are trained in mathematics and whose primary faculty appointments are in departments of mathematics. Virtually all computer instruction at the precollege level is now provided by teachers trained primarily in mathematics.

As computing competes with mathematics for students' attention, both compete with new fad courses in "computer literacy." Educators and laymen are equally confused by the many competing claims for modern technology-based education. Like rocks tossed randomly into a fast-moving stream, new courses hastily introduced into the traditional curriculum create whirlpools of conflicting eddy currents. To minimize curricular turbulence and to ensure students' smooth passage, computer scientists and mathematicians must work closely together on educational issues of common concern.

The opinions of the author stated herein do not necessarily represent the position of ACM.

The Cant of Computer Literacy

"Cant" is an old English word for insincere or meaningless talk used merely from convention or habit. Originally, it referred to the sing-song chant of street beggars during the Middle Ages.

"Computer literacy" is one of the cant phrases of our age. We hear it everywhere, from local school-board meetings to national studies. More often than not the litany of "computer literacy" is repeated out of habit rather than out of careful thought.

The advocates of computer literacy conjure images of an electronic society dominated by the information industries. Their slogan of "literacy" echoes traditional educational values, conferring the aura, but not the logic, of legitimacy.

Textbooks and courses designed for computer literacy represent at most a pale shadow of the literary and scientific disciplines they would imitate. These courses contain neither a Shakespeare nor a Newton, neither a Faulkner nor a Darwin; they convey no fundamental principles nor enduring truths. Instead, they are filled with ephemeral details whose intellectual life will barely survive the students' school years.

In truth, computer literacy is more like driver education than like calculus. It teaches students the prevailing rules of the road concerning computers: how to create and save files, how to use word processors and spread sheets, how to program in BASIC. One can be confident only that most students finishing such a course will not injure themselves or others in their first encounter with a real computer in the workplace. But such courses do not leave students well prepared for a lifetime of work in the twenty-first century.

Algorithms and Functions

Sound preparation for the future requires fundamental principles of mathematics and computer science, not the sophistry of computer literacy. It is here, at the interface between computer science and mathematics, that one can confidently build an educational foundation for the information age.

Algorithms and data structures are to computer science what functions and matrices are to mathematics. As much of the traditional school mathematics curriculum is devoted to elementary functions and matrices, so beginning courses in computing—by whatever name—should stress standard algorithms and typical data structures.

For example, as early as students study linear equations they could also learn about stacks and queues; when they move on to conic sections and quadratic equations, they could in a parallel course be investigating linked lists and binary trees. The algorithms for sorting and searching, while not part of traditional mathematics, convey the power of abstract ideas in diverse applications every bit as much as do conic sections or derivatives.

Computer languages can (and should) be studied for the concepts they represent—procedures in Pascal, recursion and lists for LISP—rather than for the syntactic details of semicolons and line numbers. They should not be undersold as mere technical devices for encoding problems for a dumb machine, nor oversold as exemplars of a new form of literacy. Computer languages are not modern equivalents of Latin or French; they do not deal in nuance and emotion, nor are they capable of persuasion, conviction, or humor. Although computer languages do represent a powerful way to solve problems, they are not a new form of literacy.

Refocusing Collegiate Mathematics

Patterns of undergraduate mathematics are changing significantly, largely in response to the needs of computer science. Topics from discrete mathematics are flavoring traditional courses throughout the curriculum. New special courses are being introduced, and new textbooks written.

But as computer scientists have special needs in mathematics, so mathematicians have special needs in computer science. The undergraduate program in mathematics needs to incorporate computer methods not only for those who will become computer scientists, but also for those who will become mathematicians, engineers, teachers, and scientists.

On the one hand, analysis of algorithms leads naturally to issues of speed and efficiency, and to the mathematics of computational complexity. On the other hand, scientific challenges of nonlinear systems, turbulent phenomena, and chaotic behavior compel new approaches to large-scale scientific computing. In either case, whether in the mathematics of computer science or in the mathematics of physical science, students who study such material soon learn how the power of computers rests on fundamental principles of both mathematics and physics.

The most important impact that the study of computer science can have on the study of mathematics is the same as that which the study of mathematics should have on the study of computer science: an appreciation of the modes of thought that characterize both fields. Rather than memorizing computer jargon in the name of a new "literacy," students must be encouraged to understand and appreciate the modes of thought that lie at the interface of mathematics and computer science.

Computer Science in the Schools

The March 1985 issue of *Communications* contains a report of the ACM Task Force on Teacher Certification. It is good to see a forceful statement that those who teach precollege computer science must have a solid background in the academic discipline of computer science. That has been the tradition in all other disciplines, including mathematics. As a target for the long run, and for large districts that can afford computer science specialists, this certification program will be very helpful.

But for the short term we must deal with the reality of 100,000 secondary school teachers of mathematics who are continually being asked to teach bits and pieces of computer science. Both for prospective teachers now in college and for currently practicing teachers, the most common pattern in districts will be to assign computer courses to teachers certified in mathematics (or perhaps in science). New certification recommendations notwithstanding, for the rest of this century the majority of high-school students will receive instruction in computer science from a teacher of mathematics.

The challenges of both precollege and collegiate education offer common cause to both the computer science and mathematics communities. As computing joins mathematics as a basic ingredient in secondary and higher education, we must move beyond computer literacy. Mathematicians and computer scientists share common objectives in this endeavor, and must work together to bring them about.

Author's present address: Lynn Arthur Steen, President, Mathematical Association of America, St. Olaf College, Northfield, MN 55057.