Published on *Inside Higher Ed* (https://www.insidehighered.com)

Why every student should study computer science (opinion)

Submitted by Robert Sedgewick on October 28, 2019 - 3:00am

Every college student needs a computer science course, and most need two or more. More and more educators are beginning to recognize this truth, but we are a long way from meeting the need.

Should we *require* all college students to take a computer science course? That is perhaps debatable. But, without question, we need to make such courses *available* to all students.

Colleges and universities offer the opportunity for any student to take as many courses as they desire in math, history, English, psychology and almost any other discipline, taught by faculty members in that discipline. Students should have the same opportunity with computer science. But at far too many institutions today -- including many of the most prestigious in the country -- students who are not computer science majors encounter severe enrollment caps, watered-down computer science for nonmajors courses or courses that just teach programming skills. They deserve better.

Many students need computer science to prepare for success later on in the curriculum. Archaeologists write programs to piece together fragments of ancient ruins. Economists apply deep learning models to financial data. Linguists write programs to study statistical properties of literary works. Physicists study computational models of the universe to analyze its origins. Musicians work with synthesized sound. Biologists seek patterns in genomes. Geologists study the evolution of landscapes. Artists work with digital images. The list goes on and on.

Programming is an intellectually satisfying experience, and certainly useful, but computer science is about much more than just programming. The understanding of what we can and cannot do with computation is arguably the most important intellectual achievement of the past century, and it has led directly to the development of the

computational infrastructure that surrounds us. The theory and the practice are interrelated in fascinating ways. Whether one thinks that the purpose of a college education is to prepare students for the workplace or to develop foundational knowledge with lifetime benefits (or both), computer science, in the 21st century, is fundamental.

Even students who will not need to program at all are likely to have important encounters with computational thinking later in life. For example, philosophers, politicians, reporters and, well, *everyone* -- not just software engineers -- must address privacy, security and ethical issues in software.

Computer science is also fertile ground for critical thinking. How might a given program or system be improved? Why might one programming language or system be more effective than another for a given application? Is a given approach a feasible way to attempt solving a given problem? Is it even possible to solve a given problem? A course or two in computer science can prepare any student to grapple effectively with such questions.

Steve Jobs once said on National Public Radio that “computer science is a liberal art.” Whether one believes that or not, the question is undeniably debatable and in the best tradition of the liberal arts! And one cannot begin to address the question without familiarity with the basics. Computer science is grounded in logic and mathematics and relevant to philosophy, the natural sciences and other liberal arts, so it belongs in the education of any liberal arts student. Just to pick one example, developments over the past century in computer science have taken logic, one of the bedrocks of the ancient liberal arts, to new levels. Computer science is not just useful. It expands the mind.

Courses for Every Student

Whatever major they might eventually choose, students nowadays know that computer science is pervasive and that they need to learn as much as they can about it. But unfortunately, opportunities to do so are limited for far too many students. Before seriously considering the idea of requirements, colleges and universities must focus on how to provide *access* to courses for all their students.

We are far from a national consensus, but an approach that has proven successful and has promise for the future is to invest in an introductory computer science sequence that teaches the important concepts and ideas in the field, as we do for economics, physics, mathematics, psychology, biology, chemistry and many other disciplines. For

example, at my institution, Princeton University, we started work on this approach in the 1980s and now are reaching over two-thirds of all the students at the university. Other institutions have seen similar results.

When starting out at Princeton, I thought about lobbying for a computer science requirement and asked one of my senior colleagues in the physics department how we might encourage students to take a course. His response was this: “If you do a good course, they will come.” This wisdom applies in spades today. A well-designed computer science course can attract the vast majority of students at any college or university nowadays -- in fact, there’s no need for a requirement.

An important reason to develop a single introductory course that everyone takes is that it makes later courses accessible to everyone, too. Students in genomics, linguistics, astrophysics, philosophy, geosciences or whatever field who need deeper background in computer science can easily get it -- as well as easily transition to computer science as a major or minor.

Perhaps the most important benefit of the approach is that it supports diversity. The typical approach of offering an accelerated curriculum to Steve Jobs wannabes and computer science for nonmajors courses to everyone else is inherently antidiversity. It sends the message to the nonmajors that they are inferior and puts them in a position where they have little chance to catch up -- when, actually, they are not so far behind.

Does this put computer science majors at a disadvantage? No. They can learn their major in depth later, as do the doctors, chemical engineers, writers, historians and everyone else. Meanwhile, they can benefit from learning something about the big picture, along with everyone else.

By putting everyone in the same course, focusing on what is important, teaching programming in the context of interesting and diverse applications across many disciplines, avoiding esoteric language details that can easily be saved for later, and mixing in historical context, theory, simple abstract machines and other material that is new to everyone, we can get all students on more or less the same playing field in one or two courses -- pretty much in the same way as we do in other disciplines.

As recently noted in *Inside Higher Ed*, there is a supply and demand shortage "on steroids" [1] for computer science faculty, with no clear solution in sight. How can a dwindling number of faculty members possibly increase the enrollments in their courses

by a factor of five or 10, in addition to everything else they must do?

Of necessity, faculty members who teaching huge computer science courses around the world have had to find ways to get the job done that are more effective and efficient than traditional methods. In recent years, it has been exciting to see scalable approaches to teaching computing on all fronts. We can replace inefficient and ineffective large live lectures with curated online videos, use modern tools to create new and better textbooks and associated online content, and develop web services to streamline assessments. Like textbooks, these materials can be shared among educational institutions, further leveraging their effectiveness. Curated videos and web services developed at one institution can be used to improve the educational experience for students at another, in the same way as textbooks. Such developments have enabled computer science professors to reach huge numbers of students more efficiently and effectively than ever before.

Should computer science be *required* of all students? Maybe. But the first step for any college or university is to commit to providing access to at least a full year of computer science for each and every student. That is what their students want and need. Modern technology can help give it to them.

Robert Sedgewick is the William O. Baker '39 Professor of Computer Science at Princeton University. He is the recipient of the Karl V. Karlstrom Outstanding Educator Award from the Association for Computing Machinery for textbooks and online materials that have educated generations of computer science students worldwide.

Section:
The Curriculum [2]
Editorial Tags:
Computer science [3]
Image Source:
Istockphoto.com/Andrew Suslov
Is this diversity newsletter?:
Disable left side advertisement?:
Is this Career Advice newsletter?:
Trending:

Links

[1] https://www.insidehighered.com/news/2018/05/09/no-clear-solution-nationwide-shortage-computer-science-professors

[2] https://www.insidehighered.com/news/news-sections/curriculum-1

[3] https://www.insidehighered.com/editorial-tags/computer-science