Eric S. Roberts
Department of Computer Science
Stanford University
Stanford, California 94305 9015 USA
eroberts@cs.stanford.edu
http://cs.stanford.edu/~eroberts/

Meeting the Challenges of Rising Enrollments

FOR MOST OF THE LAST DECADE, the number of computer science majors declined significantly at colleges and universities throughout the United States, in spite of continuing demand from industry for graduates in the field. In the last few years, however, many institutions have witnessed a dramatic turnaround in enrollment, with many more students choosing to become CS majors, or at least exploring that possibility by taking introductory computer science courses.

In the past year, enrollments rose even more quickly at many universities, including Stanford, where several of our spring quarter courses had twice as many students as they did just one year ago. Several other universities report similarly rapid growth in enrollments, to the point that the national media have noticed the resurgent interest. In June 2011, *The New York Times* reported that "enrollment in computer science programs, and degrees from them, are rising after a decade of decreases," ascribing the resurgence of interest, at least in part, to "Hollywood's portrayal of the tech world, as well as celebrity entrepreneurs like Steven P. Jobs [and Mark] Zuckerberg who make products that students use every day." [4]

CS enrollments are cyclical

To underscore the volatility in recent enrollment patterns, the Times story includes the chart in Figure 1, which tracks the fluctuation in the number of computer science majors over the past decade. This chart, however, tells only part of the story. One problem is that the Computing Research Association's annual Taulbee Survey from which this chart is derived [2] includes only research universities and therefore does not report a large fraction of the degrees produced each year. A more serious problem is that the time period covered by Figure 1 is too short to reflect important long term trends.

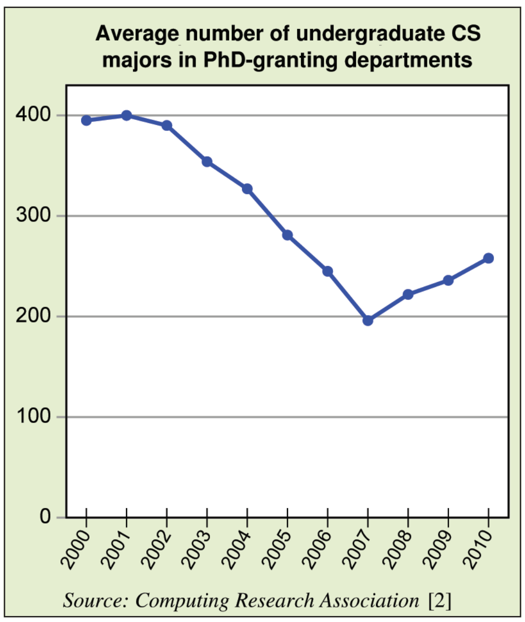

Figure 1: Taulbee Survey

Figure 2, which comes from the more comprehensive data collected by the U.S. Department of Education, offers a more complete picture. Over its relatively short history, computer science has been subject to unusually high variability in enrollment compared to most fields. As Figure 2 makes clear, computer science experienced enormous growth from 1980 to 1986 and then again from 1997 to 2003. The first of these expansions is associated with the rise of the personal computer; the second corresponds to the dot com boom that followed the commercialization of the World Wide Web. Each of these growth spurts is followed by a period of contraction, although the decline is not as steep as the climb that precedes it.

Interestingly, Figure 2 provides no evidence of the growth in majors that many universities have reported in the last few years. That fact, however, is not at all surprising. For one thing, the Department of Education reports data slowly. The most recent statistics come from 2009, which is before the resurgence was well established. For another, the number of Bachelor's degrees produced in a year is a lagging indicator. Students make their decisions about majors several years before they actually receive their degrees. Students began to turn away from computer

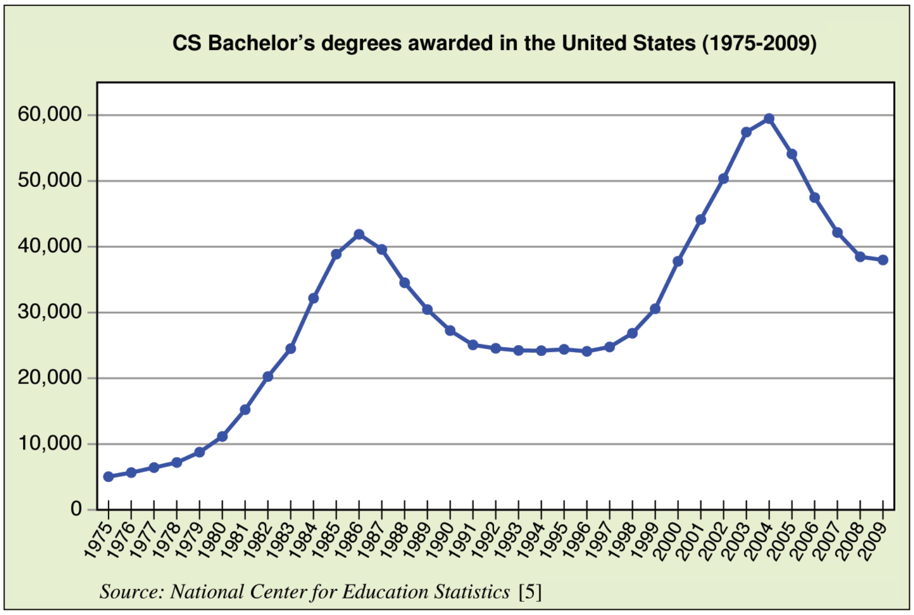

Figure 2: National Center for Education Statistics Degree Data

science majors after the dot com collapse in 2000 01, but the number of degrees continued to grow until 2004. Current evidence points unequivocally to a new period of rapid growth, but it will take time for that trend to appear in the data from the Department of Education.

Each cycle is different

One of the important lessons that computer science educators must learn from history is that enrollment in computer science follows cycles that have little or nothing to do with the structure of our courses or our curriculum. At the same time, it is equally important to recognize that the factors that lead to cyclical enrollment patterns are not always the same. The boom and bust cycles we have seen are the result of social and economic forces that change from cycle to cycle. Each cycle has an individual character, and it is important to be mindful of the differences.

As an example of how social and economic factors shape the nature of an enrollment boom, it is useful to compare the current resurgence of interest in computer science with the dot com boom of the late 1990s. In both of those periods, a flourishing computing industry provided exciting—and lucrative—opportunities for students choosing to major in computer science. In the 1990s, however, the economy as a whole was chugging along well. Today, by contrast, employment prospects outside the computing sector typically are dismal.

Such differences in the underlying economic context have a profound effect on the processes by which students decide what courses they should take. In the dot com era, students felt relatively little pressure to pursue computer science, given that there were jobs for most graduates regardless of field. Students felt free to pursue their passions, which is precisely what we encourage them to do in their college years. In those days, the passion that seemed to drive far too many computer science majors was the pursuit of wealth.

The situation today is somewhat different. Although there are some students who aspire to follow in the footsteps of Mark Zuckerberg, most of the students flocking to our introductory courses have come to bolster their credentials before they emerge into a weak economy. Most are pursuing majors in other areas but recognize—probably correctly—that programming skills will boost their chances of gaining employment in their own field. A surprising number of those students, once they get into our introductory courses, fall in love with the material and end up changing their major to computer science.

We are therefore left with a rapidly increasing stream of majors and course enrollments that have eclipsed all previous records, at least here at Stanford. As the Taulbee data in Figure 1 show, other research departments are seeing much the same thing. And while it is not yet clear to what extent this pattern of growth has spread beyond the research departments, history suggests that research universities serve as trend setters in this regard. Patterns that emerge at universities like Stanford and Harvard—both of which were covered in the *New York Times* article—are more likely to get media attention. That coverage goes on to create its own momentum that leads to a widespread revival of interest.

The need for growth

The fact that computer science enrollments are rising again is great news, not only for computer science as a discipline, but also for the economy as a whole. The collapse in the number of computer science majors that followed the dot com collapse left industry facing a critical shortage of software developers with the required training and expertise.

Occupational outlook projections in the computing sector (2008–2018)

	2008	2018	Δ
Computer and information scientists, research	28,900	35,900	24.2%
Computer programmers	426,700	414,400	-2.9%
Computer software engineers	909,600	1,204,800	32.5%
Computer support specialists	565,700	643,700	13.8%
Computer systems analysts	532,200	640,300	20.3%
Database administrators	120,400	144,700	20.2%
Network and computer systems administrators	339,500	418,400	23.2%
Network systems and data communications analysts	292,000	447,800	53.4%
All other computer specialists	209,300	236,800	13.1%
Total	**3,424,300**	**4,186,800**	**22.3%**

Source: *Bureau of Labor Statistics* [1]

Figure 3: Bureau of Labor Statistics Employment Data

That shortfall, moreover, seems likely to get worse in the years to come. Figure 3 shows the most recent projections from the Bureau of Labor Statistics on the likely growth patterns for a variety of job categories in the computing field. Between 2008 and 2018, the BLS projects the creation of 762,500 new jobs in the computing sector. Unless the number of CS majors increases, universities and colleges in the United States will produce only 380,000 computer science graduates over that period. That total is less than half the number required to cover the newly created jobs, not to mention replacing workers who retire from the field.

The risks of rapid growth

To meet the projected demand from industry, universities and colleges in the United States will have to double or triple the rate at which they produce computer science graduates. Doing so, however, is more easily said than done. There are many barriers that make it difficult for academic institutions to respond to shifts in demand in a timely way. Academic institutions change slowly, even in the best of times. At least in terms of the economy, today's times are anything but the best.

Given the current funding climate for higher education, it is difficult to imagine how most universities and colleges will be able to make the investments necessary to support increased numbers of students in computer science. In the wake of budgetary crises, public support for education is falling at both the state and the federal level. In the absence of new resources, doubling the number of students will simply double the burden on computer science faculty—an outcome that has proved to be unsustainable in the past.

Learning from our failures

One of the important lessons that our community needs to learn from history is that the "bust" phase of the enrollment cycle follows patterns that are just as idiosyncratic as those on the "boom" side. The bust in the early 2000s grew out of a decline in student demand in the wake of the dot com collapse. By contrast, the bust in the late 1980s arose instead from a failure of capacity. From 1979 to 1986, the number of computer science graduates increased by a factor of four—a growth rate even faster than Moore's Law. During that period, most departments could not grow to meet the demand, simply because there were so few candidates for faculty positions.

In the 1980s, the inability to hire new faculty made it impossible for most departments to satisfy the increased student demand. As a result, institutions were forced to discourage student interest by adopting such strategies as limiting the size of the computer science major or staffing courses with inadequately trained outsiders.

In 1982, Kent Curtis of the National Science Foundation produced a brilliant report documenting the problems facing computer science in the early 1980s. Despite the passage of nearly thirty years, much of the report is completely relevant to the situation we find ourselves in today, to the point that Curtis's report should be required reading for anyone seeking to understand the risks we face from the recent explosion in student demand. Because the original report is difficult to find, I have created a web based version and put it up on my Stanford web site. [3]

Ph.D. production is substantially higher today than it was in the early 1980s, so the problem is no longer that there is no one to hire. Unfortunately, it makes very little difference in practice whether faculty hiring is precluded by a paucity of candidates or by a lack of resources to make the necessary appointments.

Where do we go from here?

Despite the funding problems that institutions face, there are several strategies that departments can adopt to teach larger numbers of students without incurring unsustainable costs, many of which echo proposals I made during the dot com boom [7]:

1. *Develop industrial partnerships.* Industry has both the incentive and the resources to support computer science education in universities and colleges. It is, after all, industry that suffers most from a shortage of well trained graduates.
2. *Hire faculty who focus on teaching.* Increasingly, the leading universities have recognized the importance of hiring effective teachers to design and deliver introductory courses in computer science. This strategy not only increases the flow of students into computer science but can also create cost savings through economies of scale.
3. *Hire and train undergraduate TAs.* Along with many other universities, Stanford uses undergraduates as teaching assistants in the introductory courses. In our experience, these TAs are more effective than their graduate counterparts, even though they cost less to hire. Moreover, the pool of available undergraduate TAs grows along with the number of majors, thereby creating a self balancing resource. Over the years, we have published several papers describing this program [6, 8], which we believe has been central to our success.

As a nation, we cannot afford to repeat the failures of the early 1980s. As we emerge from a decade in which far too few students chose to major in computer science, it makes no sense to frustrate the renewed student enthusiasm by turning yet another generation away because of a lack of resources. The economy needs more people with computer science training, and we have a collective responsibility to prepare students for those positions. **Ir**

References

[**1**] Bureau of Labor Statistics. *Occupation Employment by Industry: 2008 18 National Employment Matrix*. U.S. Department of Labor. ftp://ftp.bls.gov/pub/special.requests/ep/ind occ.matrix/occ_pdf/occ_15 1000.pdf

[**2**] Computing Research Association. *CRA Taulbee Survey*. 2010. http://www.cra.org/resources/taulbee/

[**3**] Kent K. Curtis. Computer manpower: Is there a crisis? National Science Foundation, 1982. http://cs.stanford.edu/~eroberts/papers/Curtis ComputerManpower/

[**4**] Claire Cain Miller. Computer studies made cool, on film and now on campus. *The New York Times*, June 11, 2011, p. A1.http://www.nytimes.com/2011/06/11/technology/11computing.html

[**5**] National Center for Education Statistics. *Digest of Education Statistics*. U.S. Department of Education, 2010 (and earlier years). http://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2011015

[**6**] Stuart Reges, John McGrory, and Jeff Smith. The effective use of undergraduates to staff large introductory CS courses. *Proceedings of the Nineteenth SIGCSE Technical Symposium on Computer Science Education*, Atlanta, Georgia, February 1988. http://doi.acm.org/10.1145/52965.52971

[**7**] Eric Roberts. Conserving the seed corn: Reflections on the academic hiring crisis. *Inroads*, December 1999. http://doi.acm.org/10.1145/349316.349363

[**8**] Eric Roberts, John Lilly, and Bryan Rollins. Using undergraduates as teaching assistants in introductory programming courses: An update on the Stanford experience. *Proceedings of the Twenty sixth SIGCSE Technical Symposium on Computer Science Education*, Nashville, Tennessee, March 1995. http://doi.acm.org/10.1145/199691.199716

DOI: 10.1145/2003616.2003617