Notes on Computer Science Education -- Is it Broken?

Herbert J. Bernstein

Dowling College, Oakdale, NY 11769-1999

Is it Broken?

- First we need to decide what we are trying to do:
  - Fill classroom seats
  - Graduate “CS majors”
  - Teach appreciation of computers
  - Teach use of computers
  - Teach design of solutions to problems using computers
  - Teach creation of solutions to problems using computers
  - Make the world a better place
  - …
- Probably all of the above and more

Is it Broken?

- Something is broken
  - The NSF thinks so (e.g. CPATH NSF 06-608)
  - Students think so (enrollments are down nation-wide) L. Tucci, “College Students Continue to Shun Computer Science,” CIO News, 8 June 2005
  - Industry thinks so (outsourcing keeps rising) “Outsourcing Grows While Computer Science Degrees Shrink”, Computerworld, 5 May 2006
  - Many peers in other fields think so (they rely on COTS and “amateurs” for their programming) -- personal observation and …

Is it Broken?

- Something is broken (continued)

  "There are some who argue that it does not matter if computer science, as a discipline, withers a bit. They say fields that rely on computer science -- which is to say, virtually all fields -- will develop their own expertise in-house, so to speak, as scientists and engineers accumulate the skills they need, almost ad hoc, as they do their research"

  -- Cornelia Dean, reporting in the Science Times, The New York Times, Tuesday, April 17, 2007, pp F1-F2.

Is Everything Broken?

- No, many of us are doing a good job at many things:
  - Teaching use of computers
  - Teaching appreciation of computers
  - Teaching students how to teach students about computers
  - Teaching students how to manage people who solve problems with computers
- **But not enough of us are teaching enough students how to solve problems with computers**

What is Broken?

Personal Opinion:

- We are not guiding our students to the skills they need
  - Problem-solving skills
  - Communications skills
  - Programming skills
- We are not guiding our students to the values they need
  - Work habits
  - Integrity
  - Ethics

How Do We Fix It?

From the movie City Slickers:

> "Curly: Do you know what the secret to life is? Holds up his index finger. This. One thing. Just one thing. You stick to that and everything else don't mean nuthin.
>
> "Mitch: That's great, but what's the one thing?
>
> "Curly: That's what you gotta figure out."

Some of us have to make solving this problem our "one thing"

How Do We Fix It?

- Require our students to acquire the skills they need
  - Problem-solving skills
  - Communications skills
  - Programming skills
- Require our students display the values they need
  - Work habits
  - Integrity
  - Ethics
- Teach student to be professionals by requiring them to work as professionals

What We are Trying at Dowling

Slow, careful sequence in programming

JavaScript, Java, C/C++

Programming in many courses

Project-Oriented Computer Science

- One 12-credit full academic year course
- Multiple team-based projects
- Programming in multiple languages
- Student seminars

Involve students in multi-disciplinary research

Introductory Sequence

Programming First, but done slowly

First course uses Javascript

- Simple
- Minimal use of objects and typing
- Immediate feedback
- Useful, even for non-majors

Second course uses Java

Third course is data structures

Advantage: Majors have time to learn to program, non-majors can cope with Javascript

Disadvantages: Sequence take a year and a half and Javascript is not well supported

Programming in Many Courses

Students need a lot of programming practice to gain fluency and confidence

Programming assignments help students appreciate practical issues

Helps students to get over phobias about learning new languages.

Project-Oriented Computer Science (POCS)

Spend the junior year gaining the skills needed to do "real work", to do multiple projects, to work in teams

Prepares students for senior year internships

Works well in the face of low enrollments

Important issue: some students resist the heavy workload, because it is not the norm in college these days

Multidisciplinary Research

- Active players in externally funded grants (DOE, NIH, NSF)
  - Learn to solve problems that matter
  - Learn to prepare presentations of their work
  - Learn to present at “real” meetings (both oral presentations and posters)
- We do bioinformatics
  - 2-3 students a year get to go to national or international meetings and present their work
  - Other students get to be co-authors, to put our their work as open-source code, to deal with users world-wide.

Role of Open Source

We use a local open source server and sourceforge for both course project work and student research

Trains students in open source collaborations, both internally and with external developers

Makes work from earlier cohorts easily available to later cohorts

Encourages students to be very open about their progress

Helps students to build their portfolios

Problem: Makes the game designers and others who want to get rich on proprietary packages uncomfortable

Where Do We go From Here?

We keep doing different, useful things

Some of us are going to push our students very hard to be well-trained designers and implementers of computer-based solutions to problems, not because it is the only way to teach CS, but because for some teachers and some students it works

Advantages to Multidisciplinary Research Programs

- Activities involve freshman to seniors
  - Yes, freshman can do useful work
- Projects have continuity and students teach one another
  - Running such a program is less work than you might think and help tremendously in establishing good work habits
- Students hang out together and get to know one another
  - They learn to compete in an open, constructive manner
- Students help other students get jobs
  - It really works