

# Large Introductory Courses in Research Computer Science Departments

Moderator:

**David G. Kay**, Lecturer in Computer Science, Department of Information and Computer Science, University of California, Irvine, CA 92697–3425 — [kay@uci.edu](mailto:kay@uci.edu)

Panelists:

**Jacobo Carrasquel**, Lecturer in Computer Science, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15123 — [jxc@cs.cmu.edu](mailto:jxc@cs.cmu.edu)

**Michael J. Clancy**, Senior Lecturer, Computer Science Division, University of California, Berkeley, CA 94720 — [clancy@cs.berkeley.edu](mailto:clancy@cs.berkeley.edu)

**Eric Roberts**, Professor and Associate Chair for Education, Department of Computer Science, Stanford University, Stanford, CA 94305 — [eroberts@cs.stanford.edu](mailto:eroberts@cs.stanford.edu)

**Joseph Zachary**, Associate Professor and Director of Educational Programs, Department of Computer Science, University of Utah, Salt Lake City, UT 87112 — [zachary@cs.utah.edu](mailto:zachary@cs.utah.edu)

Introductory courses offered by computer science departments in research universities have several characteristics that distinguish them from courses with similar content at other institutions:

- Often the class sizes are large, ranging even to hundreds of students per class (see [1]).
- The course content changes frequently, but often not in a direction that parallels the research interests of faculty.
- Often the main group of faculty in the department are not significantly rewarded for time they devote to introductory education, which can decrease their involvement in those courses.

This panel will explore strategies, policies, and procedures used by different research departments to address these issues. Topics covered will include the use of non-research faculty to teach and manage introductory courses and curricula; levels of support for hardware, software, section leaders, lab assistants, graders, clerical and technical staff; teaching loads; and procedures for making curricular decisions.

The panelists all have significant responsibility for introductory courses in their departments. Each of their departments has taken specific steps to address the conflict between research demands and introductory teaching in computer science.

## **Jacobo Carrasquel**

At Carnegie Mellon we have a group of non-tenure faculty that is in charge of teaching all the introductory CS courses. This group has a director, an associate director, and five full-time lecturers. Currently we teach five differ-

ent 100-level (first-year) courses for several departments and schools. Our largest “clients” are the School of CS, the College of Engineering, and the College of Science.

The courses are taught every semester (15 weeks) and we have approximately 400 students every semester. In addition to the 100-level courses, we also teach a 200-level course in data structures for non-CS majors.

Our courses do not include a lecture for the entire class. Each course is structured around “recitations,” which are groups of 25 students. Each lecturer is in charge of three recitations. Recitations meet three times a week for 50 minutes each (or twice a week for 70 minutes). On the average, each lecturer must spend 10 contact hours and three office hours.

All the lecturers agree on a common set of topics to be taught during the semester, on the language and platform to be used, and on the final exams. Each lecturer decides on the outline, lab assignments, other exams, and textbooks for his or her recitations.

We don’t have traditional TAs. We have help with some of the grading and office hours from undergraduates that we call “Class Assistants”. Each recitation has one CA assigned to it; the CA will attend class and spend time in the labs working with students.

## **Michael J. Clancy**

The U.C. Berkeley CS faculty includes two lower-division specialists, myself and Brian Harvey. We are both classified as “lecturers” rather than “professors;” the Lecturer job description does not include research responsibilities, and thus involves a higher annual teaching load (somewhat less than double that of a professor). We both have the equivalent of tenure. Brian’s duties mainly involve teaching—he runs three sections per year of 200 or

more students—while mine include substantial administrative responsibilities. Both of us have also taught upper-division courses in the past.

Certain aspects of “second-class citizenship” are inherent in the Lecturer position; for instance, we are not allowed to vote on tenure cases, and nontenured lecturers are due for a smaller cost-of-living adjustment this year than assistant professors. However, our department administrations have gone out of their way to reduce the impact of these institutional hassles on us. I sit on all the committees relevant to lower-division instruction (undergraduate study, computer needs and resources, scheduling), and Brian and I both have substantial input in decisions about curricular evolution and resource allocation.

Regarding other aspects of Berkeley's support for computer science instruction, here are some data. Our student-teacher ratio in all courses is around 50 students to 20 hours of TA. Lower-division courses get 10 minutes per student per week of graders; upper-division courses, slightly less. Our student/computer ratio is between 10 and 15 to 1. Three rooms of UNIX workstations are dedicated to the lower-division CS major course sequence, and more than half the available hours of a room of Macintoshes are allocated to our service course for non-majors. A significant portion of the budget for instructional support staff is dedicated to the lower-division courses.

### **Eric Roberts**

As at most institutions, the enrollment patterns in computer science courses at Stanford place considerable pressure on teaching resources. Although the computer science department represents only 2% of the university faculty as a whole, the department is responsible for about 6% of the units taught. Enrollment is also heavily skewed toward introductory and service courses. The typical department in the School of Engineering teaches a majority of its units to its own majors; in computer science, almost 80% of the units are taken by students majoring in other departments throughout the university.

Meeting this high level of demand using regular faculty would place a disproportionate burden on the department that would make it difficult to maintain our position as the top-ranked research department in the United States. As a result, Stanford has relied for almost twenty years on a separate teaching faculty to supply much of the instruction. This group, which is called the Educational Affairs Division, currently consists of one faculty member and six lecturers. The faculty member serves as Associate Chair of the department and is appointed with the rank of Professor (Teaching), which is a special faculty line at Stanford with full faculty status but off the tenure clock. Appointments consist of renewable five-year contracts. Lecturers are typically appointed for terms of between two and three years and are considered for promotion to Senior Lecturer after seven years.

Lecturers are responsible for two types of courses:

1. Introductory courses, which are interpreted as extending beyond the CS1/CS2 level to include the standard sophomore course.

2. Service courses, which consist of practically oriented courses targeted primarily to our master's program and students from local companies who are enrolled via the Stanford Industrial Television Network (SITN).

The distance education provided by the SITN program is an important aspect of our teaching work, not only because it provides a valuable educational service to the community, but also because it changes the economics of the teaching program. Because of the SITN revenue from service courses, the lecturers bring in more money to the department than they cost—a fact that makes it much easier to justify the program to senior university officials.

Lecturers are hired entirely on the basis of quality of teaching; no research is expected. As a result, most lecturers receive very high teaching evaluations, even in large, high-workload courses where such ratings are more difficult to achieve. The effective teaching load beginning in 1996-97 is four courses over three quarters. Teaching support in the form of TAs and undergraduate section leaders is available to courses taught by lecturers at the same level as those taught by faculty. Each lecturer's preferences are taken strongly into account when course assignments are made.

### **Joseph Zachary**

The Department of Computer Science at the University of Utah has 19 regular professors and two non-tenure-track “clinical” professors. There are approximately 300 undergraduate and 90 graduate students studying in the department at any given time.

The two clinical faculty members run every aspect of the undergraduate program: they handle admissions, teach almost all of the freshman-level courses, advise students, schedule courses, instigate most curriculum changes, and provide input on hardware and software purchases. Clinical faculty teach five or six courses per year, whereas regular faculty teach three. (Utah is on the quarter system, which means that the teaching load for regular faculty is one course per quarter.) Release from some of the teaching load for a clinical faculty member can be arranged to allow time to work on such projects as textbook writing and educational software development.

For the most part, clinical faculty teach the courses that are left over after the regular faculty have chosen. Lately those courses have been the CS1/CS2 equivalents, sophomore/junior-level programming languages and hardware design courses, and senior/graduate-level compiler and programming languages courses.

Faculty (whether regular or clinical) are typically supported with one graduate or undergraduate teaching assistant for every 30–40 students enrolled in a class. The teaching assistants do most of the grading and staff the computer laboratories.

### **Reference**

- [1] Kay, D. G., Carrasquel, J., Clancy, M. J., Roberts, E., and Zachary, J. Managing large introductory courses (panel presentation). *Proceedings of the 28th SIGCSE Technical Symposium on Computer Science Education*, 1997, pp. 386–387.