

January 21, 2005  
Teaching Tip  
Teaching Large Classes

Good afternoon everyone. Every once in a while I receive some information about teaching and learning from colleagues that is worthwhile to pass along to BGSU faculty. Such is the case with the following posting below that looks at an innovative way to challenge the best students in a class without leaving those at the low end behind. It is from an article by Dawn Levy about award winning computer science professor Eric Roberts' work with student contests. It appeared in the December 3, 2004 issue of the Stanford Report. <http://news-service.stanford.edu/> Reprinted with permission.

In general, the article gives us some ideas about how to teach to a large group of students whose abilities and knowledge vary greatly. Although the specifics of how to do this are somewhat lacking in this overview article, it is worth reading just to see possibilities of thinking beyond such notions as grading on the curve and teaching only to certain groups of students. As Roberts sees it, we need to give opportunities to all of our students—even in a large class.

Have a nice weekend.

---

#### CONTESTS MOTIVATE TOP STUDENTS IN LARGE CLASSES

Eric Roberts tackles neglected issue of motivating elite students through recognition, camaraderie, teaching opportunities.

Roberts provides ample opportunities for his students to earn extra-credit through contests that usually entice about 10 percent of a given class to enter. Work is judged for categories including "best algorithm" and "best aesthetics." Past winners have programmed adventure games and created computer animations.

Eric Roberts is the principal architect of what was for many years the largest course at Stanford-Computer Science 106A, an introductory programming class with an enrollment that waxes and wanes with the NASDAQ. In a fat year, 1,000 students may enroll, with more than 400 students in a single class. How professors can encourage top scholars in large classes was the topic of Roberts' "Award-Winning Teachers on Teaching" talk Nov. 18 in Building 460.

"How do you manage in a large course not completely snowing the people on the low end of the scale or boring silly the people on the high end of the scale?" asked Roberts, the John A. and Cynthia Fry Gunn University Fellow in Undergraduate Education.

Only 6 percent of Roberts' introductory students end up majoring in computer science. Many major in other engineering disciplines, and almost 20 percent major in social

sciences or humanities. "It's a wide spectrum, and I wanted to encourage those people to be in that class," Roberts said. "It was one of the signature aspects of the Stanford curriculum that we tried to keep an introductory computer science sequence that really did have broad appeal."

Success in such classes means supporting students every step of the way so they can do well, Roberts said. It also means setting a high bar. "One of the difficulties in a large class is if you decide that you're going to curve it rigidly, you've forced it into a mode where many people are going to be unhappy," he said. Roberts favors a grading system that rewards those who meet clearly delineated objectives-no matter how many students meet those objectives. "If everybody does enough work to get an A, then everybody will in fact get an A."

That's a big incentive to do well. Further, superlative work in Roberts' classes has earned A+ and even A++ marks. (The latter designates work that "exceeds all expectations," according to a jury of section leaders, teaching assistants and the professor.)

Top students also have the opportunity to gain teaching experience after the class ends. Owing to economic necessity and a dearth of graduate students willing to assist teaching introductory computer courses, the teaching assistants (TAs) in such courses at most universities are undergraduates.

"We decided at Stanford to make a virtue of necessity and really train those students to be wonderful as teachers," Roberts said. Undergraduate TAs also provide "stepping stone role models" that help increase the number of underrepresented minorities in computer science.

Programming is one of the most varied intellectual activities in terms of productivity and ability, Roberts said. "The difference between this person who's sort of good and that person who's really great is extraordinary."

A 1968 study of working programmers showed 20 to 1 variations in productivity-how much code a person could generate-among individuals with the same levels of education and experience. The best programmers also tended to be the fastest and to have the fewest bugs. "You see the same [enormous variability] in classrooms," Roberts said.

Move over, Ed McMahon...

Roberts said he can't gear the class to top students without risking losing those on the bottom. Instead, he leverages the features of large classes to encourage excellence. And that means-drumroll, please-lots of contests that provide extra credit. Students can enter three contests per quarter in his CS 106 A and B classes. Only large classes have enough students to make contests feasible, as only about 10 percent of the class usually enters. A large class may have 40 to 60 entrants. Winners rise from the obscurity of a large class.

Judged by an army of section leaders, programs can win for such categories as "best

algorithm" or "best aesthetics." Some contests have explored the limited world of Karel the robot, who can turn left but not right, forcing students to program three left turns to make a right when Karel runs a maze. One winning program calculated the weakest and strongest countries in a game of Risk to determine who could attack whom. Another winner created an animation of IBM's Deep Blue supercomputer beating chess champion Garry Kasparov in 1997. Yet another winner created an adventure game using text-based commands to have players find magic wands and potions.

Prizes provide recognition (certificates for first-place, runner-up and honorable mention contestants, and classroom presentations of winning entries), camaraderie (a group dinner for winners at Roberts' house the following quarter), perks (offers of letters of recommendation from Roberts) and glory (an automatic 100 percent score on the final exam for the first-prize winner of any contest).

The contests have opened up new worlds for some students. When a former English major in Roberts' Computer Science 105 course won a contest, she became so fired up that she got an undergraduate degree in symbolic systems and a master's degree in computer science. She went on to become employee number nine at Google.

---

---