Using Undergraduates as Teaching Assistants in Introductory Programming Courses: An Update on the Stanford Experience

Eric Roberts, John Lilly, and Bryan Rollins
Department of Computer Science
Stanford University

ABSTRACT

The introductory computer science classes at Stanford University have a combined annual enrollment of over 1000 students. Teaching introductory programming to a population of this size requires significant instructional support to ensure that students receive the individual attention most beginning programmers need. This paper describes the approach that has evolved at Stanford for the introductory computer science courses, which is based on the extensive use of advanced undergraduates to teach sections of the introductory course. In our experience, using undergraduates as part of the teaching staff has created an effective learning environment for both the students in the classes and the undergraduate section leaders themselves.

1. INTRODUCTION

For more than a decade, the Computer Science Department at Stanford University has used undergraduate students to supplement the instructional staff in the introductory programming courses. The typical introductory course is assigned one lecturer, one graduate teaching assistant, and a large number of undergraduate course assistants, called "section leaders." The lecturer is responsible for presenting the class material and designing assignments and exams. The graduate TA helps the lecturer with course content and administration. The section leaders conduct weekly small-group discussion sections, grade assignments, and assist students working in the computer cluster.

In 1988, Stuart Reges, then the director of Stanford's computer science education program, presented a paper at the SIGCSE conference describing Stanford's initial experiences with the section-leading program [Reges88]. In that paper, Reges discusses the following advantages of the program:

- Undergraduate section leaders are less expensive than traditional graduate TAs.
- Undergraduate section leaders, having more recently been in the same position, are better able to establish a rapport with introductory students.

- Undergraduates are more familiar with the Stanford computing facilities and curriculum than most graduate students.

While these arguments continue to apply, our six additional years of experience have demonstrated several other positive features of the section-leading program:

- Having the opportunity to teach new students is of enormous benefit to the section leader as well. In the process of explaining concepts to each new class, the section leader's own understanding of those concepts grows dramatically. Similarly, encouraging students to use good programming style usually improves the section leader's programming style. Thus, the section-leading program becomes an integral part of the section leader's own undergraduate education.
- The section-leading program has a salutary effect on the entire undergraduate department, even though only about 25 percent of CS majors participate in the program, which also includes students from many other fields. Their shared experience creates a strong connection among the section leaders, which gives rise to many activities that benefit the community as a whole.
- The section leaders, occupying as they do a position of significant trust and responsibility, become role models for younger students. That modeling is all the more effective because the section-leader's role is one that an introductory student might assume, not in ten or twenty years, but in one or two. The fact that the path to becoming a section leader is well-marked gives many undergraduates a powerful incentive to succeed.
- The section-leading program serves as a training ground for future lecturers. Seven of the 13 people hired as departmental lecturers over the past decade were formerly Stanford undergraduates involved in the section-leading program. By participating in the program, section leaders develop excellent teaching skills that serve them well in more advanced teaching positions.

In short, we are more convinced than ever that using undergraduates as section leaders has significant pedagogical advantages, both for the introductory classes and for the program as a whole. Thus, we thought it was appropriate to update the 1988 Reges paper and offer a more detailed analysis of the Stanford approach.
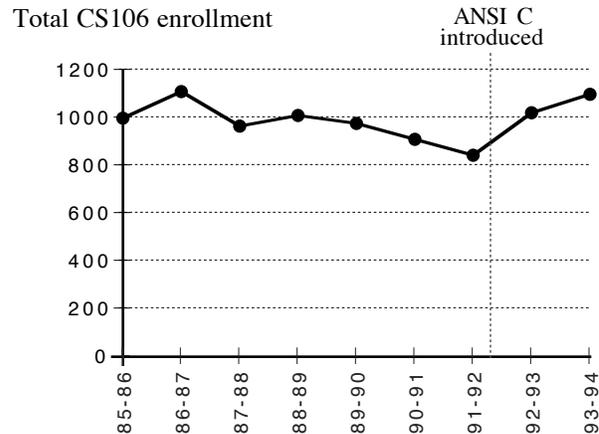
2. INTRODUCTORY CS COURSES AT STANFORD

In order to appreciate the structure of the section-leading program and how it fits into the curriculum, it is necessary to know something about the Stanford computer science program as a whole and, in particular, about the scale at which it is conducted. At present, we estimate that 75 percent of Stanford undergraduates take at least one introductory programming course offered by the Computer Science Department. To appeal to students with a wide range of interests and prior experience, the department offers several different introductory courses, as illustrated in this diagram:
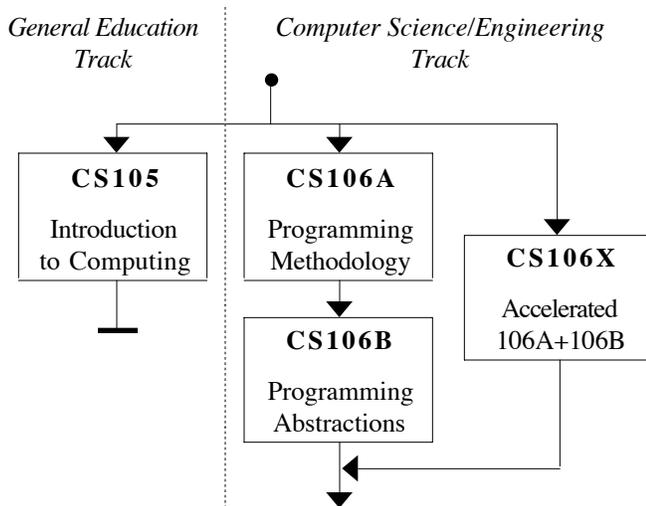
CS105 is intended for nontechnical students, most of whom are seeking to satisfy Stanford's undergraduate requirement in technology and applied science. The CS106 courses offer a more traditional introduction to programming that follows reasonably closely the CS1/CS2 curriculum outlined in Curriculum 78 [Austing79]. The CS106A/CS106B sequence covers this material in two quarters and is designed for students with little or no programming experience. CS106X covers the same material in a single quarter and is designed for students with more extensive programming backgrounds.[1]

As at most universities, enrollments in Stanford's introductory programming courses skyrocketed in the late 1970s and early 1980s, increasing by a factor of three in the decade between 1975 and 1985. Since 1985, enrollments in the introductory courses have remained relatively constant, as illustrated by the following graph showing the total annual enrollment in all sections of CS106:

---

[1] Although the material is largely review for many advanced students in CS106X, we encourage even highly advanced students to begin their computer science study at the CS106X level. Most students—even those with years of programming experience in high school—have weak software engineering skills, which improve considerably when they take this course.

From 1986 to 1991, enrollments in CS106 registered a downward trend. This decline, however, was reversed when we changed the programming language from Pascal to C at the end of the 1991–92 year [Roberts93].

It is also interesting to note that most undergraduates—even those who are majoring or intend to major in a nontechnical field—tend to take one of the CS106 courses rather than the nontechnically oriented CS105. Typically, only about 300 students take CS105 each year. By contrast, the combined enrollment in CS106A and CS106X (covering the two paths through the CS106 sequence) is approximately three times that size.

The CS106 series is popular because

- The courses are regarded as highly practical, particularly now that they use ANSI C.
- CS106 fulfills an Engineering School requirement, while CS105 does not.
- CS106 has an excellent reputation in the undergraduate community for high-quality teaching.

Course evaluation statistics support the good reputation of CS106. The following table summarizes the student ratings for all CS106 courses in 1993–94 in comparison to all courses in the Engineering School:

	CS106 courses	All engineering courses
1 (excellent)	63 %	44 %
2 (very good)	31 %	34 %
3 (good)	5 %	14 %
4 (fair)	1 %	6 %
5 (poor)	0 %	2 %

As these statistics indicate, 94 percent of the respondents rate CS106A as very good or excellent. In 1994, one of the undergraduate newspapers named CS106A as one of the five best courses at Stanford.

The reputation of the CS106 courses is further enhanced by the section-leading program, which supports all the CS106 courses but not CS105, whose design is more conventional in that all teaching assistance is supplied by

graduate TAs.[2] The section leaders act as ambassadors for the CS106 program in the undergraduate community and encourage other students to take those courses.

3. RESPONDING TO LARGE ENROLLMENTS

The combined enrollment for all sections of CS106 in 1993-94 was 1112. Teaching introductory programming at this scale places considerable demand on the resources of the Computer Science Department. Although computer science professors represent only two percent of the total university faculty, computer science courses account for five percent of the total student units, with much of the additional demand for instruction coming from the introductory courses. This imbalance between faculty size and enrollments means that the Computer Science Department carries two and a half times the average instructional load per faculty member—a level that cannot be sustained without jeopardizing Stanford's preeminent position as a computer science research institution.

To reduce the teaching load to a level more consistent with that in the rest of the Engineering School, Stanford appoints lecturers to supplement the regular faculty. Lecturers are chosen entirely on the basis of demonstrated teaching ability and are not expected to conduct research. The positions are renewable but not eligible for tenure. Currently, the department employs five lecturers, two of whom work part-time, so the total lecturer strength is the equivalent of 4.25 full-time positions. In addition, the Associate Chair for Education, a faculty-rank position within the department currently held by Eric Roberts, oversees the entire teaching program.

The lecturers, as talented as they are, would not be able to do such a good job with these large courses without assistance. Teaching introductory programming effectively requires that students be able to interact with someone who understands the material, particularly when questions arise in the context of a programming laboratory or a homework assignment. Introductory programming courses tend to require a large number of student contact hours, far more than would be possible if the lecturer had sole responsibility for this aspect of the class as well. At Stanford, most of the teaching support required for the introductory courses is supplied by the section-leading program, which is described in the next section.

4. SECTION LEADERS

In a typical quarter, the Computer Science Department hires approximately 50 section leaders to staff the CS106 courses. At the beginning of each quarter, every section leader is assigned to a section consisting of approximately ten students. Throughout the quarter, those students "belong" to that section leader, in that the section leader is responsible for going over material with them in a weekly recitation section, grading their programs, and being their link to the rest of the class. For instance, if students feel

lectures are covering the course material too quickly or that particular topics have been a little confusing, the section leader can spend extra time on those topics during section that week and make the lecturer aware of the section's concerns. This feedback mechanism is extremely important for large lecture courses in which it is difficult for the lecturer to get an accurate reading of the class as a whole. This structure provides all students with a small-group environment in which they are not merely one student in a sea of 300 others.

For two hours a week, each section leader holds what we call "helper hours" in the campus computer cluster, where they are available to help any student who is working on a CS106 assignment. These hours are distributed among section leaders to cover approximately 70 hours per week, with additional coverage during times of unusually heavy use. We believe that providing this type of readily available assistance is extremely important for computer science in particular. Students often discover that they are having conceptual problems only when they reach an impasse at the computer. Having a staff member on call to explain concepts, answer questions, and help the student discover the problem not only makes an enormous difference in the level of student understanding but also reduces the level of frustration associated with the class. In essence, the existence of a large section-leader community makes it possible for Stanford students to enjoy the benefits of closed laboratories on a round-the-clock basis.

The section leaders represent a diverse cross-section of the Stanford undergraduate population. In educational status, they range from sophomores to students spending an additional year to obtain both an undergraduate and a master's degree. Although many of the section leaders are computer science majors, a significant number come from other fields, including biology, psychology, economics, political science, and feminist studies. More than a quarter of the section leaders are women, which is larger than the fraction of women in the major. Providing role models for a wide variety of students helps make the CS106 course more accessible, and we work hard to maintain diversity in the program.

5. SUPPORT STRUCTURE

Although the section-leader program has been extremely successful at Stanford, much of that success is due to the support structure that has grown up around the program. Undergraduates often lack the depth of experience that a graduate student might bring to a TA position; they therefore need training, assistance, and administrative support to be optimally effective in their role as members of the teaching staff.

The CS198 course

The most important aspect of the support structure for the section-leader program is the CS198 course—Teaching of Computer Science. In their first quarter with the program, section leaders enroll in CS198, which is designed to prepare them to be more effective teachers. New section leaders receive university credit for CS198 but are not paid during their first quarter of work. Thereafter, section leaders

---

[2] Until five years ago, CS105 also used undergraduate section leaders to provide course assistance. Since that time, however, CS105 and CS106 have used different programming languages, which makes it difficult to support both courses with a single unified program.

are paid on an hourly basis, which usually works out to 10–15 hours a week.

Because new section leaders must begin teaching section in the second week of the quarter, CS198 is designed to cover as much information as early in the term as possible. The course is presented in a workshop rather than a lecture format and meets twice a week at the beginning of quarter, once a week in the middle, and not at all towards the end. There are approximately ten workshops in all, including "Grading," "Debugging," "Teaching Styles," "Creativity in Teaching," and "How to Present Material."

The workshop format allows the new section leaders to develop not only as individuals, but also as a group. Since we do not believe that a single formula exists for teaching, the workshops allow new section leaders to explore their own ideas and discuss them with their peers. In the process, they learn as much, if not more, from the other section leaders as they do from the workshop leaders.

In addition, the workshops allow new section leaders to get to know one another and establish connections and friendships with other section leaders. Each new section leader is paired with an older, more experienced one as part of a "buddy system" that helps integrate new people into the program.

The CS198 coordinators

Like any organization with a staff of 50, the section-leader program requires administrative coordination. At Stanford, that responsibility rests with two graduate students, both former section leaders, who are hired to serve as coordinators for the program. Because a large part of the job consists of running the CS198 class, these people are usually called the "CS198 coordinators"; their charge, however, is actually much broader and includes the following tasks:

- *Managing the administrative mechanics of the program.* The coordinators are responsible for determining how many sections are needed, scheduling classrooms, assigning section leaders to specific courses and times, and distributing students among the individual sections. The coordinators also identify and resolve problems as they arise.

- *Hiring new section leaders.* Each quarter, the coordinators solicit applications from undergraduates who want to become section leaders. Competition for these positions is stiff; in a typical quarter, only one candidate out of four is selected. The selection is based on two principal criteria: thorough knowledge of the relevant course material and teaching effectiveness. The coordinators assess teaching effectiveness largely on the basis of an interview in which the candidate gives a short presentation on one of the topics likely to come up in section.

- *Conducting the CS198 course.* As discussed in the preceding section, every new section leader takes CS198. The coordinators are responsible for planning the curriculum and leading the workshops.

- *Offering supplementary workshops on essential topics.* The coordinators conduct occasional review sessions on topics that are likely to cause confusion, such as strings, memory allocation, and abstract data types. These workshops are open to all section leaders and provide continuing education for experienced section leaders, who are not enrolled in CS198. These workshops have proven valuable in easing the transition caused by the change in programming language, because they show section leaders who were introduced to certain concepts in Pascal how to teach those same concepts in C.

- *Running weekly staff meetings.* The staff members for all three CS106 courses meet together once a week during the term, so that lecturers can review what's going on in each class and what questions to expect from students in connection with the current assignment. Section leaders need to keep track of the assignments for all three classes so that they can respond effectively to the questions they get from students in the computer cluster.

- *Fostering communication.* The success of the section-leading program depends on good communication. Just as students depend on section leaders for help and information, the entire course staff relies on the coordinators to provide a network that links together the various individuals associated with the program.

The coordinators report to the Associate Chair for Education and interact with the entire staff of lecturers. Particularly as the program has matured, however, the coordinators maintain a significant level of autonomy, which allows the program to run smoothly most of the time without intensive faculty supervision.

Grading systems

Many section leaders find grading to be the most difficult and problematic aspect of their job. While grading is never easy, undergraduate section leaders face a special challenge for the following reasons:

- Section leaders do not have the experience of a faculty member or a graduate TA. Although they understand the course material well, they may be uncertain about how to evaluate a student who chooses an unconventional approach.

- Grading can establish an adversarial dynamic between the section leader and the student, which sometimes makes it harder to establish the individual connection that makes the section-leading program so successful.

- Because they are much closer to the student in age and academic status, section leaders sometimes have problems with students who want to argue about their grades. A faculty member quickly learns how to handle the student who insists that a B+ should really have been an A–; a section leader has less experience and less authority to resist that pressure.

To make it easier for section leaders to evaluate student work without jeopardizing their relationship with the class, we have developed a grading system for CS106 that has the following characteristics:

1. *The course staff develops detailed grading criteria for each assignment.* Before each assignment is evaluated,

the lecturer, TA, and section leaders work together to prepare a comprehensive set of guidelines for grading the assignment. After looking at a sample of the submitted assignments, the members of the staff discuss what they are looking for in terms of a solution, what alternative strategies might be possible, and how much weight should be assigned to errors of various types. Section leaders can then refer to the grading criteria as they read and evaluate the assignments.

2. *Grading is interactive.* When a student turns in a program, the section leader looks it over, writes comments on the paper, and assigns a preliminary grade based on the specific grading criteria established by the staff as a whole. The section leader then schedules an appointment with the student in which they talk about the program, running it on a computer if it helps to illustrate specific points about the coding. This dialogue turns out to be invaluable. The student receives verbal feedback that is far more helpful than the written comments but also has a chance to explain to the section leader the strategy behind a particular solution. This discussion helps the section leader evaluate how well the student understands the critical concepts.

3. *Grades are assigned on a qualitative scale that is extremely coarse.* After discussing all the positive and negative aspects of the assignment with the student, the section leader assigns the paper to one of several categories set by the lecturer. Although the grading systems used by different instructors vary to some extent, one of the common approaches is to assign grades from the following set:

++ This grade must be authorized by the instructor and is reserved for the absolutely wonderful assignment that appears at most only once or twice a quarter—the sort of program that "makes you weep" when you see it.

+ This grade is used for programs that exceed the expectations of the course. Assignments that merit a + must incorporate extra features beyond the minimal requirements or get the job done in a particularly elegant way.

$\checkmark+$ This grade designates a submission that satisfies all the assigned requirements—a job well done.

$\checkmark$ This grade indicates that the program meets the general requirements for the assignment, possibly with a few minor problems.

$\checkmark-$ This grade signifies that the program has problems serious enough to fall short of the assigned requirements.

− Programs that receive a − grade have extremely serious problems but nonetheless demonstrate some effort and understanding.

−− This grade is used for programs that show little effort and do not represent passing work.

Because almost all the grades are $\checkmark+$, $\checkmark$, or $\checkmark-$, the psychological importance of grading is reduced, which makes it possible for students to concentrate more on the feedback they receive during the interactive grading session.

The section-leader community
Because interpersonal ties are always an important part of any support structure, the section-leader program tries to promote a community atmosphere that is open and inclusive. The coordinators are continually involved in sponsoring an environment in which section leaders can get to know the TAs, the lecturers, and each other. Section leaders get together for such activities as barbecues, basketball, four-square, and laser tag. By encouraging communication and interaction among section leaders, these informal activities contribute as much to the quality of the program as do the meetings and workshops.

The section leaders also play an important role that is not listed as part of any job description: to serve as public representatives of the computer science teaching community. Section leaders are recognized throughout the campus as people who understand computer science and programming. It is not unusual for a section leader to wander into a dormitory-based computer cluster and help four or five people with their programs. They occasionally meet students for lunch to go over problems with a particular assignment. Most section leaders have had discussions with students about the pros and cons of the computer science major or what courses would be most valuable for them to take. This informal presence has had a significant impact on the undergraduate population, to the point that new computer science majors often have a good idea of what the department is like before they actually declare their major.

6. CONCLUSIONS
In our experience, using undergraduates as part of the teaching process has proven to be an extremely effective way to provide teaching support for introductory computer science classes. The system not only benefits students in the class but also has a profoundly positive effect on the section leaders involved in the program and the computer science community as a whole.

REFERENCES
[Austing79] Richard Austing, Bruce Barnes, Della Bonnette, Gerald Engel, and Gordon Stokes, "Curriculum '78: Recommendations for the undergraduate program in computer science," *Communications of the ACM,* March 1979.
[Reges88] Stuart Reges, John McGrory, and Jeff Smith, "The effective use of undergraduates to staff large introductory CS courses," *SIGCSE Bulletin,* February 1988.
[Roberts93] Eric S. Roberts, "Using C in CS1: Evaluating the Stanford experience," *SIGCSE Bulletin,* March 1993.