

## CS 106B — General Information

---

### Professor: Eric Roberts

E-mail: [eroberts@cs.stanford.edu](mailto:eroberts@cs.stanford.edu)  
Offices: Gates 202  
Phone: 723-3642  
Drop-in hours: Tuesdays, 9:30–11:30 A.M.

### TA: Michael Gummelt

E-mail: [mgummelt@stanford.edu](mailto:mgummelt@stanford.edu)  
Office: Gates 160  
Phone: 723-6083  
Office hours: Mondays, 10:00–11:00 A.M.  
Thursdays, 2:30–3:30 P.M.

### Lectures

Lectures are scheduled for Monday, Wednesday, and Friday at 3:15 P.M. in Hewlett 200. The schedule for individual lectures is given on the accompanying calendar handout. CS 106B lectures are videotaped only in the spring, so this quarter's lectures will not be available on the web.

### Discussion sections

In addition to lecture, you **must** sign up for a weekly 50-minute section. In order to take CS 106B, you must sign up for a section before 5:00 P.M. Sunday, January 15 using the signup form on the web at <http://cs198.stanford.edu/section/>. **Important note: Do not use Axxess to sign up for sections; section enrollments made through Axxess are simply ignored.**

### Units

If you are an undergraduate, you are required to take CS 106B for 5 units of credit. If you are a graduate student, you may take CS 106B for 3 or 4 units if it is necessary for you to reduce your units for administrative reasons. Taking the course for reduced units does not imply any change in the course requirements. **Important note: The default on Axxess is to sign you up for 3 units; make sure that you change this entry to 5.**

### Drop/add deadlines

You may not add or drop a course after 5:00 P.M. on Friday, January 27. The deadline for withdrawing from a course or for changing the grading status is 5:00 P.M. on Friday, March 2.

### Web page

The web page for CS 106B is <http://www.stanford.edu/class/cs106b/>. All course materials will be posted on this site, so be sure to check it frequently.

## Text and handouts

The text for this course is a completely new draft of a forthcoming textbook entitled *Programming Abstractions in C++*. The first 13 chapters of the draft are available as a reader from the Stanford Bookstore. We will distribute the remaining six chapters, along with the appendix and index, as handouts later in the quarter. The draft textbook is available on the class web site, so that you can refer to it online. You will, however, almost certainly want a copy because you will need it for the open-book exams.

As a supplement to the text, we will distribute additional material in the form of class handouts. After class, any extra copies of the handouts will be placed in the “Handout Hangout” bins in the entryway to the Gates B-wing between Gates 182 and 188. The handouts are also available in PDF<sup>®</sup> format on the CS 106B web site. If you miss a handout in class, you can print your own copy from the web.

## Examinations

The midterm examination will be administered at two different times on **Thursday, February 9: from 3:15 to 5:15 P.M. and from 7:00 to 9:00 P.M.** The final examination will also be offered at two times: **Monday, March 19, and Wednesday, March 21, on both days running from 12:15–3:15 P.M.** The examinations themselves are written so that you should be able to complete them in less than the allotted time: the midterm is designed to take one hour and the final is designed to take 100 minutes. All examinations are open-book, and you may use any notes or materials from the class.

## Programming assignments

CS 106B requires six programming assignments, which are due on the dates given in the syllabus. Except for Assignment #6 (which is due at the very end of the quarter), your section leader will grade each of your assignments and then discuss how well you have done (and you can improve on future assignments) in an interactive, one-on-one session. Assignments are graded separately for functionality and style using the following scale:

- ++ An absolutely fantastic submission of the sort that will only come along a few times during the quarter. To ensure that this score is given only rarely, any grade of ++ must be approved by the instructor and TA. Since your section leader would almost certainly want to show off any assignment worthy of a ++, this review process should not be too cumbersome.
- + A submission that exceeds our standard expectation for the assignment. The program must reflect additional work beyond the requirements or get the job done in a particularly elegant way.
- ✓+ A submission that satisfies all the requirements for the assignment—a job well done.
- ✓ A submission that meets the requirements for the assignment, possibly with a few small problems.
- ✓– A submission that has problems serious enough to fall short of the requirements for the assignment.
- A submission that has extremely serious problems, but nonetheless shows some effort and understanding.
- A submission that shows little effort and does not represent passing work.

From past experience, we expect most grades to be ✓+ and ✓. Dividing the grades into categories means that your section leader can spend more time talking about what you need to learn from the assignment and not have to worry about justifying each point.

For each assignment, you must make an appointment with your section leader for an interactive-grading session. Your section leader will explain in section how to schedule these sessions and go over the grading process in more detail.

### **Late policy**

Each of the assignments is due at 5:00 P.M. on the day specified in the syllabus. Assignments are submitted electronically using the procedure described in the first assignment. **Important note: all assignments are due at 5:00 P.M. sharp on the dates indicated on the assignment handout. Assignments submitted after 5:00 will be considered late.**

Because each of you will probably come upon some time during the quarter where so much work piles up that you need a little extra time, every student begins the quarter with two free “late days.” To avoid any ambiguity, a “day” is defined as a class day. Thus, if your assignment was due on Friday but turned in the following Monday, that assignment would be one day late. (Note: Assignment #6, which is due at the end of the quarter after classes have finished, has its own late-day policy in which “days” revert to calendar days.) After your late days for the quarter are exhausted, programs are assessed a late penalty of one category point per late day used (a ✓+ turns into a ✓, and so forth). Late days are valuable, and it pays to keep some around for the harder assignments toward the end of the quarter. In all cases, assignments must be turned in within a calendar week of their published due date.

In special circumstances (such as extended medical problems or other emergencies), extensions may be granted beyond the late days. To request an extension, send e-mail to the TA (Michael Gummelt, [mgummelt@stanford.edu](mailto:mgummelt@stanford.edu)) no later than 24 hours before the program is due. Only the TA is authorized to approve such extensions. In particular, do not ask your section leader.

### **Contests**

As shown on the calendar handout, there are three contests scheduled at different points during the term. The point of these contests is to give you a chance to show creativity and initiative beyond what is formally required by the course. Rules for each contest will be distributed in class when they are announced. The grand prize in each contest is a perfect score on one of the graded components of the course, typically the final exam.

Beyond the contests there are additional incentives to encourage your participation. First, every serious entry gets you one virtual ticket in a random drawing for a special grand prize at the end of the quarter. Second, winning a runner-up prize or honorable mention in a contest or having an assignment judged as a ++ candidate gives you additional chances in the random drawing, which will take place at the beginning of the review for the final exam. Finally, as a way of getting the textbook into a more polished form, being the first person to report a bug in the draft or to make a substantive suggestion that I decide to implement will receive chances in the final drawing. These chances are not earned linearly, but rather follow a pattern of quadratic decay (you’ll learn more about

linear and quadratic efficiency later in the quarter). The first bug you find earns you one chance. To get the second chance, you have to report two more. The third chance requires reporting three more bugs, for a total of six. The threshold for earning  $N$  chances in the final drawing is therefore the  $N^{\text{th}}$  triangle number.

## Grading

For as long as anyone can remember, the most important component of the final grade in the CS 106 courses has been the programming assignments, which have typically counted for approximately half the final grade. Even so, one of the biggest complaints we hear from students is that the assignments don't count for enough relative to the exams. Many students feel that exams are not a particularly good measure of one's knowledge. If, after all, someone can implement the assignments effectively, why should it matter how that person performs on an examination taken under arbitrary time constraints without the aid of a computer?

Although I'm entirely sympathetic with this argument in theory, there is a problem. Computer science courses—here at Stanford as well as at most other institutions—have historically been marked by an intolerably high number of Honor Code cases, often representing more than half of the cases that arise at Stanford in a year. Given the severity of the problem, it has always seemed necessary to use exams as a check to make sure that students have in fact learned the material. Someone who copies their assignments from someone else may do very well on those assignments (assuming we don't catch it), but will in all probability do poorly on the midterm and the final.

Three years ago, I tried an experiment in CS 106B that I have repeated ever since. If Honor Code violations do not seem to be a problem—and the guidelines in Handout #3 should provide some guidance here—we'll give the assignments even greater weight than we traditionally have. If, however, cheating rears its ugly head, we'll shift that weight to the exams.

Here, then, are the weights for the different components of the course:

Programming assignments	60% minus 5% for each Honor Code case reported
Final examination	15% plus 5% for each Honor Code case reported
Midterm examination	15%
Section participation	10%

For example, if no Honor Code cases at all come up this quarter in CS 106B, the programming assignments will count for 60% of the grade, and the final will be worth only 15%, making it the same as the midterm. If, however, two Honor Code cases are reported, the assignments will be worth only 50% and the final will count for 25%, which is pretty much what we've traditionally done. If more Honor Code cases come up, the weight on the assignments will continue to decrease, but in no case will drop below 15%.

There is one further clarification that we need to make. The only violations that count are suspected Honor Code violations reported by the course staff. If *students* in CS 106B report an Honor Code violation (as you are supposed to do, after all), we won't count that report against the class as a whole. This policy, therefore, gives everyone a strong collective incentive to maintain academic integrity.

### **Computer facilities**

Programming assignments can be implemented on either Macintosh or Windows PC computers, using either XCode (on the Macintosh) or Visual Studio C++ (on the PC). Information on these programming environments will be provided in separate handouts. Many of you will do your work in the Tresidder LaIR computer cluster, which is staffed most evenings with section leaders who can help if you get stuck.