CS 106B — General Information

Lecturer: Chris Gregg

E-mail:	`cgregg@stanford.edu`
Office:	Gates 191
Office hours:	TBA

Head TA: Jason Chen

E-mail:	`cheson@stanford.edu`
Office hours:	TBA

Overview – What is CS 106B?

CS 106B, "Programming Abstractions" is the follow-on course to CS 106A. It is taught in C++, although students are only expected to know some Java (or other object oriented language) before enrolling in the course. That said, if you are taking CS 106B without taking CS 106A, you should probably look at the Course Placement handout from the website to see if you are in an appropriate course.

Class web page

The web page for CS 106B is located at `http://www.stanford.edu/class/cs106b/`

You can also reach that webpage via the simpler URL: `http://cs106b.stanford.edu/`

You should regularly check the class web site for handouts, announcements and other information, including the most up-to-the-date information on assignments and errata. **Please note that the class web page will have links to essential class materials including electronic copies of class handouts and assignment files.**

Discussion sections

In addition to lecture, you must also sign up for a weekly 50-minute section. In order to take CS 106B, you must sign up for a section by 3:30P.M. Tuesday, June 27th. The signup form is available on the web at the URL `http://cs198.stanford.edu/section/`. After a matching process, your section assignments will be e-mailed out to you. Sections begin the first week of classes, because the summer quarter is so short. Note that you should only sign up for sections at the URL indicted previously (you should not sign-up for sections on Axess). If for some reason you miss the signup deadlines, you should wait

until after June 27th at 3:30pm and go to the CS198 site and pick the "Add/Switch Section" link.

Section leaders and course helpers

CS106B provides extensive assistance for students. Section Leaders and Course Helpers are available from Sunday through Thursday evenings each week in Old Union to help with assignments. Check the web site **http://cs198.stanford.edu/** and click on the "Helper Schedule" link for the latest schedule of Helper Hours.

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 enroll in CS 106B for 3 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.

Texts and handouts

CS106B has one required textbook *Programming Abstractions in C++* by Eric Roberts. In addition to this textbook, we will also distribute additional material in the form of class handouts. Class handouts will be available electronically in PDF format on the CS 106B web site. If you prefer printed handouts, you can print a copy from the web.

Programming assignments

As you can see from the syllabus, there will be seven assignments (Assignment 1 – Assignment 7). The assignments will become slightly more difficult and require more time as the quarter progresses. Thus, the later assignments will be weighed slightly more than the earlier ones. Except for Assignment #7 (which is due at the very end of the quarter), each assignment is graded during an interactive, one-on-one session with your section leader, who rates it according to 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 is "perfect" or exceeds our standard expectation for the assignment. To receive this grade, a program often reflects additional work beyond the requirements or gets the job done in a particularly elegant way.

√+   A submission that satisfies all the requirements for the assignment, showing solid funtionality as well as good style. It reflects 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. The overall goal is to maximize the learning experience in doing the assignments, and we have found the "bucket" grading system to work much better for programming assignments than assigning numeric grades from a pedagogical perspective over many quarters of experience.

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 **12 noon** on the dates specified in the syllabus. The program code for your assignments must be submitted electronically as described in a separate handout. Anything that comes in after noon 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 three free "late credits." "Late credits" comprise a single calendar day. If, for instance an assignment is due on a Thursday at noon, the student may turn it on Friday by noon, and they will use one late credit. After the late credits are exhausted, programs that come in late (up to a maximum of three class days) will be assessed a late penalty of one grade "bucket" per day (e.g., a √+ turns into a √, and so forth). Assignments received later than three class days following the due date will not be graded. The interactive-grading session with your section leader must be scheduled within two weeks of the due date. **Note that no assignments will be accepted after the last day of classes (August 17th).**

You should think of these free "late credits" as extensions you have been granted ahead of time, and use them when you might have otherwise tried to ask for an extension. As a result, getting an extension beyond the two free "late credits" will generally not be granted. In *very special* circumstances (primarily extended medical problems or other emergencies), extensions may be granted beyond the late days. All extension requests must be directed to the head TA no later than 24 hours before the program is due. Only the head TA will be able to approve extensions. In particular, do not ask your section leader.

Examinations

The midterm examination will be held over a two-day period, during class on **Wednesday, July 19th and Thursday, July 20th**. Given that the exam is during regularly scheduled class time, you should be prepared to take it during that time (i.e., there will not be an alternate exam time).

The final examination is scheduled for **Saturday, August 19th from 8:30am-11:30am**. For a variety of reasons (including university policy), **there will be no alternate time for the final exam**. Please make sure that you can attend the final exam at the specified time before enrolling in the class.

Examinations will be closed-book, and you may use a one page back-and-front page of notes, as well as a reference sheet that will be provided. You cannot use electronic devices of any type (i.e. portable computers, phones, etc).

Grading

Final grades for the course will be determined using the following weights:

- 40%   Programming assignments (weighted toward the later assignments)
- 30%   Final examination
- 20%   Midterm examination
- 10%   Section participation