

CS 106B
Chris Gregg, Lecturer
Nick Troccoli, Head TA

Handout #1
April 2, 2018

CS 106B — General Information

Based on a handout by Mehran Sahami and Chris Piech. Updated 4/23/18 for new 6PM HW due time.

Lecturer: Chris Gregg

E-mail: cgregg@stanford.edu
Office: Gates 191
Office hours: See website

Head TA: Nick Troccoli

E-mail: troccoli@stanford.edu
Office hours: See website

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://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 between 5P.M. on Thursday, April 5 and 5P.M. Sunday, April 8th. The signup form is available after logging in to <http://cs198.stanford.edu>. Section signups are **not** first-come, first-serve. After a matching process, your section assignments will be viewable starting at 5P.M. Tuesday, April 10 on the same website. Sections begin the second week of classes. Note that you should only sign up for sections at the URL indicted previously (you should not sign-up for sections on Axxess). If for some reason you miss the signup deadline, you should wait until after April 10th at 5P.M., at which point a late signup link will be available on <http://cs198.stanford.edu>.

As a side note, some of the programming assignments in this class will be done individually and for others you will have the option to work in pairs (more on that later). If you do decide to work in a pair for those assignments, you may only pair with someone in the same section time/location as you. Please keep this in mind when signing up.

Course Support

CS106B provides extensive assistance for students. Section Leaders and Course Helpers are available from Sunday through Thursday evenings each week in Tresidder Union (first floor eating area) to help with assignments. The LaIR is open 7PM-11PM. Nick and Chris also have their own office hours - please see the course website for times.

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

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 functionality 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.

Working in Pairs

Most of the assignments in this course must be completed on an individual basis, but some of them allow you to *optionally* work in a pair with a partner. Each assignment will specify if it is to be done individually or allows working in pairs. Note that you are not required to work with a partner on assignments that allow it, but you are encouraged to do so. Working in pairs can improve student learning by giving you someone to talk to when you are stuck, or by letting you see a different way of approaching the same problem. You can also change pairings between assignments. In other words, you don't have to keep the same pairing for every assignment that allows pairs (and you can even choose to do some in pairs and others individually).

If you choose to work with a partner, you must pair with another student who is currently taking the course and is in your section. If you have a friend you want to work with, request the same section or request a section swap if necessary. Students auditing or sitting in on the course may not work in a pair with a student who is taking the course. No person who is not currently enrolled in the course may be part of any pair.

If you submit an assignment as a pair, each of you are expected to make a significant contribution toward solving that assignment. You should not claim to be part of a pair submission if you did not contribute significantly to help solve that program. **If you submit an assignment as a pair, you should make ONE submission and make sure that the names of both members of the pair are listed in the comments of the**

solution. Both members of a pair will receive the same grade and do their interactive grading session together.

It goes without saying that regardless of pairs, every student is still responsible for learning all course material. All exams are completed individually. More details about working in pairs will be discussed in class and additional information will be posted on the class web site. Please make sure that you follow its guidelines.

Late policy

Each of the assignments is due at **6PM PST** 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 two free "late days." "Late days" are class days, not actual days. For instance, if an assignment is due Thursday at 6PM, 1 late day would give you until the next class (e.g. Friday) at 6PM to submit without penalty. 2 late days would give you until the next, next class (e.g. Monday) at 6PM to submit without penalty. After the late days 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 class day (e.g., a $\checkmark+$ turns into a \checkmark , 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 (June 6th).**

If you are working in a pair and turn in an assignment late, both members of the pair will be assessed "late days". For example, if you turn in your assignment as a pair one day late, then both members of the pair each incur one "late day." If you are out of free "late days", but your partner isn't, then your assignment grade is penalized one grade "bucket", but your partner would simply use one of his/her free "late days" (and thus not be penalized one grade "bucket"). So you can think of "late days" being measured per student, and we apply any penalties *individually* for submissions that are made in pairs. Note: you cannot transfer free late days to your partner.

You should think of these free "late days" 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 free "late days" 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 on **Thursday, May 3rd, 6-8pm**. There will be alternate times available for legitimate excuses (e.g., symphony practice, or another class).

The final examination is scheduled for **Friday, June 8th, 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 administered on a computer, 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) other than the computer on which you are taking the exam, which may only be used to administer the exam (e.g. you cannot view any notes).

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

Students with Documented Disabilities

Students who may need academic accommodations should contact the Office of Accessible Education, who can prepare written accommodations that should be provided to Nick and Chris. Students should contact the OAE as soon as possible since notice is needed to prepare accommodations.