

CS 106X
Chris Gregg, Lecturer
Aaron Broder, Head TA

Handout #1
January 9th, 2017

CS 106X — General Information

Based on a handout by Mehran Sahami and Chris Piech

Lecturer: Chris Gregg

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

Head TA: Aaron Broder

E-mail: abroder@cs.stanford.edu
Office hours: TBA

Overview – What is CS 106X?

CS 106X is an accelerated version of CS 106B. We will cover roughly the same topics as 106B, but we will cover them more quickly and/or in more depth. We will cover some different and more advanced topics than 106B, as well. The assignments will be more difficult than CS 106B.

Given the paragraph above, why would you want to take 106X instead of 106B? Here are a few reasons:

1. The class is smaller. It looks like we will have roughly fifty students in the course this quarter. Because of this, both Chris and Aaron will be able to get to know you better, and we can give you more individual attention than we might be able to in a 300+ student 106B course.
2. You came to Stanford and want to be seriously challenged. If you already consider yourself a good programmer, then you will be at home in 106X, and you'll be able to refine your skills. The assignments will be long but rewarding to write.
3. We can dig deeply into the material, and there will be lots of opportunities for you to ask questions, and to have discussions about C++, Data Structures, programming in general, and anything else you want to talk about.

The caveat to the great things about 106X is that we expect a lot out of you – complaints about the course being too difficult or the assignments being too long will fall on deaf ears, and if you are concerned that you can't handle the material, you might want to consider CS 106B (also a challenging course!) instead. But, you will be proud that you made it through 106X by the end of the quarter if you stick with it.

Class web page

The web page for CS 106X is located at <http://www.stanford.edu/class/cs106x/>

You can also reach that webpage via the simpler URL: <http://cs106x.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 106X, you must sign up for a section between 5:00P.M. TBA and TBA. The sign-up form will be 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 by TBA. Sections begin the second week of classes (i.e., next week). Note that you should only sign up for sections at the URL indicted previously (you should not sign-up for sections on Axxess).

Section leaders and course helpers

CS106X 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 106X for 5 units of credit. If you are a graduate student, you may enroll in CS 106X 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

CS106X 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 106X 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 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.

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 days." "Late days" are class days, not actual days (i.e. from Monday to

Wednesday is one late day). 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 day (e.g., a $\sqrt{+}$ turns into a $\sqrt{}$, 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 late days may not be used on the last assignment and no assignments will be accepted after the last day of classes (March 17th).**

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 two 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 a two-hour test administered **outside of class from 7:00pm-9:00pm on Tuesday, February 21st.** If you have a conflict with this time, and absolutely cannot make the regularly scheduled midterm, you must send a request by electronic mail to Chris or Aaron by 5:00pm on Friday, February 10th to arrange an alternate exam time. Any alternate midterm exam will be within at most one day (earlier or later) than the regular exam time, so make sure you are available in that time window if you cannot make the regular exam.

The final examination is scheduled for **Monday, March 20th 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.

All examinations are open-book (class course reader and textbook only), and you may use any notes, handouts, or materials from the class, but 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