CS106B
Spring 2012

Handout #00
April 2, 2012

Course Information

Instructor

Keith Schwarz (htiek@cs.stanford.edu)
Office: Gates 178
Office Phone: (650) 723-4350
Office Hours: Tuesday 2:00PM – 4:00PM

Head TA

Zach Galant (galant@cs.stanford.edu)
Office: Gates 160
Office Hours: Monday/Wednesday 11:00AM – Noon
Thursday 2:00PM – 4:00PM

Website

The course website is http://cs106b.stanford.edu and it's loaded with resources for this course. There, you'll find all the handouts for this course, lecture slides, lecture code, assignment starter code, software links, and much more. I would suggest periodically polling the website to stay abreast of any important developments in the course.

Lectures

Mondays, Wednesdays, and Fridays, 10:00AM – 10:50AM in NVIDIA Auditorium.

Discussion Sessions

In addition to lecture, you must also sign up for a weekly 50-minute section. Section signups are handled online, and section signups run from Thursday, April 5 at 5:00PM to Sunday, April 8 at 5:00PM at http://cs198.stanford.edu/section. After a matching process, your section assignments will be e-mailed out to you by Tuesday, April 10. Sections begin the second week of classes (i.e., next week). This link is also available on the CS106B web page.

Although Axess lists discussion sections for this course, you should **not** enroll in your section through Axess and should instead use the above link.

Section Leaders

In addition to the instructor and head TA, there will be an army of course helpers and section leaders here to help you out. If you'd like help on any of the assignments, feel free to stop by the Tresidder LaIR between 6PM and midnight, Sundays through Thursdays. For an up-to-date schedule of helper hours, you can view the most current helper schedule by going to http://cs198.stanford.edu and clicking the “Helper Schedule” link. Again, this link is available on the CS106B web page.

Units

If you are an undergraduate, you must enroll for five units. If you are a graduate student, you may enroll for either three or five units, depending on what best fits into your schedule. Regardless of how many units you are enrolled for, the course content and requirements will be the same.

Prerequisites

CS106B assumes that you have a familiarity with programming at the level of CS106A or the AP Computer Science exam. If you already have comparable programming experience, but no official classroom experience, that's perfectly fine. Handout #02 contains placement information that you may find useful in deciding whether CS106B is right for you. As always, feel free to get in touch with us if you have any questions.

CS106L

CS106B is taught in C++, but is not designed as an in-depth introduction to the C++ programming language. We will be using our own custom libraries in place of many of the standard libraries, and will not explore some of the more nuanced portions of the language.

If you're interested in learning more about standard C++ and seeing just how much firepower the language has to offer, you may want to consider additionally signing up for CS106L. CS106L is a one-unit companion course to CS106B that is dedicated purely to exploring just how much mileage you can get out of the C++ programming language. CS106L is being taught by veteran SL and C++ wizard Dawson Zhou, and it promises to be a wonderful addition to the material covered in CS106B.

CS106L is **not** a replacement for the standard CS106B discussion section. If you enroll in CS106L, you should still sign up for a discussion section.

Email

We will use email as a primary means for distributing announcements and notifying you when your assignments has been graded, so having an email account is a requirement for this course. If you don't already have an email account, you can get one at Stanford through LaIR.

Information on obtaining an account is available at the Tresidder computer cluster, from your Resident Computer Consultant, or via the web at http://email.stanford.edu

Readings

The required reading for this course is Eric Roberts' excellent text ***Programming Abstractions in C++***, which has been newly updated and revised. It's available as a two-part course reader from the Stanford bookstore. Additionally, we will have a PDF copy of the course reader available on the course website.

Computers

In CS106B, we support PC and Mac, but currently do not have support for Linux. If you are using a PC, you will need to obtain a copy of Visual C++ 2008 Edition (details in a later handout). On the Mac, you can use Xcode (again, details to follow in a later handout).

Assignments

There will be **seven** programming assignments over the course of the quarter, each of which will give you a chance to play around with the material from lecture and build progressively more impressive pieces of software. The assignments will become 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 on both functionality (does the program work?) and style (how elegant is your code?):

++ 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 ***the start of class*** on the dates specified in the syllabus. The program code for your assignments must be submitted electronically as described in a separate handout. All assignments are due at 10:00A.M. sharp on the dates indicated on the assignment handout. Anything that comes in after 10:00A.M. 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 (i.e. from Monday to Wednesday counts as 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 ✓+ turns into a ✓, and so forth).

Assignments received later than three class days following the due date will **not** be graded, so please make an effort to get the assignments completed on-time. 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, as it comes due during the finals period normally reserved for this course. More specifically, **we will not accept any late submissions for the final assignment**, even if you have remaining late days.

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, extensions 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, Zach Galant, no later than 24 hours before the program is due. **Only Zach will be able to approve extensions**. In particular, please do not ask your section leader for an extension, since they cannot grant you one.

Grading

In addition to the seven programming assignments, there will be two midterm exams. The first midterm will be held on **Thursday, May 3** from **7:00PM – 10:00PM**, location TBA. The second exam will be held on **Thursday, May 31** from **7:00PM – 10:00PM**, location TBA. If you have a conflict that will prevent you from taking either of the exams, we'd be happy to try to find an alternate time. However, you must let us know about this no later than **two weeks in advance** so that we have adequate time to schedule another exam room and time.

Overall, your grade for this course will be determined as

Programming Assignments:	**55%**
First Midterm:	**20%**
Second Midterm:	**20%**
Section Participation:	**5%**