

Course Information

Instructor	Keith Schwarz (htiek@cs.stanford.edu) Office Hours: Tuesdays, 2:15PM – 4:15PM in Gates 219.
Head TA	Anton Apostolatos (antonaf@stanford.edu) Office Hours: Wednesdays, 2:00PM – 4:00PM in the Huang basement.
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, 11:30 – 12:20 in Hewlett 200. Lectures will not be recorded this quarter – sorry about that!
Discussion Sessions	<p>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, January 12 at 5:00PM to Sunday, January 15 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, January 17th. Sections begin the second week of classes. This link is also available on the CS106B web page.</p> <p>Although Axess lists discussion sections for this course, we don't look at Axess section enrollments when assigning sections. Even if you're enrolled in a section through Axess, you need to sign up through our system as well to make sure that someone will be grading your assignments!</p>
Section Leaders	In addition to the instructor and head TA, there will be a group 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 the LaIR or CLaIR between 6PM and midnight, Sundays through Thursdays, on the first floor of Tresidder Union. 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 (this is CS department policy). 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.
CS106S	CS106S is an optional, one-unit course to CS106B that explores applications of the material to social good – health care, civics, education, etc. It's designed as an add-on to CS106B rather than a replacement for section, so if you do enroll, make sure you also sign up for a CS106B discussion section.

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 from the Stanford bookstore and there are some copies on reserve in the engineering library.

Computers

We support all major platforms (Windows, Mac, and Linux) in CS106B. We will provide handouts describing how to get a C++ development environment set up on these platforms later this week. You can also use any of Stanford's cluster computers for this course.

Assignments

There will be *eight* 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 the final assignment (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 expectations 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

A few of the assignments in this course must be completed on an individual basis, but the majority 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 your 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 partners between assignments. In other words, you don't have to keep the same partner for every assignment that allows pairs (and you can even choose to do some in pairs and other 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 one 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 **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 11:30AM sharp on the dates indicated on the assignment handout. Anything that comes in after 11:30AM 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. Monday to Wednesday counts as one late day). No assignments may be submitted more than three class periods late. If you submit an assignment late and have used your late days, we'll still count it (provided it's fewer than three class periods late), but we will assess a flat 1% penalty to your overall grade, so it is still absolutely worth turning in an assignment late rather than not at all.

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, Anton Apostolatos, no later than 24 hours before the program is due. **Only Anton can approve extensions**. In particular, please do not ask your section leader for an extension, since they don't have the authority to grant you one.

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.

Grading

In addition to the seven programming assignments, there will be a midterm exam and a final exam. The midterm will be held on *Tuesday, February 21* from *7:00PM – 10:00PM*, location TBA. The final exam will be held on *Monday, March 20* from *8:30AM – 11:30AM*, location TBA. Note that we will *not* offer any alternate final exam times except for students with documented OAE accommodations, so you must not enroll in this class if you cannot make the final exam.

Overall, your grade for this course will be determined as

Programming Assignments:	35%
Midterm Exam:	25%
Final Exam:	35%
Section Participation:	5%