

CS106X Course Information

Instructor: Jerry Cain
E-Mail: jerry@cs.stanford.edu
Cell phone: (415) 205-2242
Office phone: (650) 725-8597
Office: Gates 192
Office hours: Mondays, 2:30 p.m. – 4:15 p.m., and by appointment

Lectures: MWF 1:15 – 2:05 p.m.
Gates B03 (and available via scpd.stanford.edu, details TBA)

Prereqs: CS106X is the more advanced of the two courses teaching basic programming abstractions and algorithms. CS106X is designed as an alternative to the more sensibly paced CS106B, because many students—self-taught hackers, exceptionally strong CS106A students, and AP Java graduates who got a 5 on the AP exam without trying too hard—prefer a more intense treatment in the company of other superstars. Students have a good time in CS106X because they're brave, resourceful, and well prepared.

AP Java and CS106A are all about basic programming practices—expressions, control idioms, decomposition, algorithmic thinking, class design, object orientation, simple inheritance, and basic client use of arrays, lists, and maps. CS106X teaches advanced programming and abstraction techniques, worrying first on C++ language mechanics and eventually focusing on topics such as recursion, C++ lists and maps, and the implementation techniques used to build complicated, dynamic data structures.

Sections: In addition to our weekly lectures, you'll also attend a weekly discussion section. The person leading your particular discussion section will be the one grading all of your assignments. There are several discussion sections to choose from. Take a moment to visit <http://cs198.stanford.edu/section> anytime between Thursday, January 10th at 5:00 p.m. and Monday, January 14th at 5:00 p.m. and state your preferred section times. Come Sunday evening, our computers will work their magic and come up with a master section assignment that will hopefully suit everybody. SCPD students need to visit the web site as well, for no reason other than to register for the televised section, which airs on Wednesday afternoons at 3:15 p.m.

- CS106L:** You may have noticed what appears to be a companion lab course, numbered CS106L. It is certainly related to CS106X, but it is completely optional, and you needn't sign up for it if you don't want to or don't have time. CS106L meets on Mondays and Wednesday at 4:15 and spends time on advanced features of the C++ language that we just don't have time for in CS106X. In particular, you'll learn the ins and outs of C++'s I/O system, and you'll learn all about the template algorithms and data structures. CS106L is being taught by a veteran section leader who knows C++ just about as well as anyone, so it's definitely worth your time if you expect to be coding in C++ beyond this quarter.
- Readings:** The class textbook is the course reader *Programming Abstractions in C++* by Eric Roberts (with C++ edits by Julie Zelenski). The course reader is available at the Bookstore (or will be very soon if it isn't already). In addition to the text, this course will also distribute a reasonable amount of material in the form of class handouts. After class, extra copies of the handouts will be placed in the handout bins on the first floor of the Gates Building in the B wing (in case you happen to lose a handout or miss class). In addition, electronic copies of all the handouts in PDF format will be available on the CS106X web site, so that you can print your own copies if the handout bin is empty.
- Software:** Programming assignments can be written on either Macintosh or Windows PC computers, using either XCode (on the Macintosh) or Visual Studio C++ (on the PC). More information on these two programming environments will be provided in separate handouts.
- Mailing List:** There is a class mailing list that will be used for important announcements that just can't wait until lecture. All students enrolled in CS106X are automatically subscribed to the `cs106x-students@lists` list. The list server is in touch with Axxess, so if you're signed up for the course, you're probably on the mailing list. Please make it a point to sign up as soon as possible, since I tend to send a good number of announcements out during the first week or two, and I don't want you to miss out.
- Programs:** There are seven or eight programming assignments, and it's possible I'll throw in a written problem for color. The assignments are serious projects, and they get more and more difficult as we cover the more advanced material. The only way to learn programming is to work at it, so expect to spend lots of time in front of a computer. Your assignments are graded interactively in a one-on-one session with your section leader. The distribution and due dates for each assignment are subject to change. I'll be sure to alert you well in advance if I plan on moving the due dates around.

Exams: There will be one 120-minute exam during the quarter and a comprehensive 3-hour exam given during the final exam period. The two exam dates are as follows:

Midterm:	Tuesday, February 19 th	7:00 - 9:00 p.m.
Final Exam:	Friday, March 21 st	8:30 - 11:30 a.m.

Because the midterm is being administered at night, I'm happy to accommodate anyone who can't make it then. If you need to take the (open note, open handout, open course reader) exam at a different time, then email me at least a week prior and we'll schedule a separate time. You should expect to take the exam sometime earlier in the day.

Very few of you should have any issues with the final exam time, since that time slot is dedicated to classes meeting MWF at 1:15 p.m. If you need to take the exam at another time, then you will need to take the exam on Friday, March 21st at 3:30 p.m.

Grading: Your final grade will be computed as follows:

Programs	35 %
Midterm	25 %
Final	40 %

To receive a passing grade, you must complete satisfactory work in all areas. In particular, if you do not pass my final exam, you will not pass my class regardless of your performance on the assignments.

Late policy: The pace of this course makes it difficult for students to catch up once they have fallen behind, so I encourage you to submit all of your assignments on time. Of course, we're all busy people, so I'm happy to accept late work, provided you keep up with the material. Here's how I handle lateness: You get three free late days, and you consume one late day any time you hand in work between 1 second and 24 hours late. You can use at most one late day on any particular assignment. Once you've used your three free late days up, they're gone and you can't get any more. You can still hand in your work, but I levy a 10% penalty per assignment for each late day you use beyond the three free ones. My recommendation: plan to get the work done on time, and use a late day if it doesn't come together by the deadline.

Incompletes: I will only grant incompletes in the event of an illness or a family emergency that makes it impossible for a student to finish his work during the regular academic quarter. In particular, you must complete all

assignments except the last one and take the midterm exam in order to qualify. Unless your circumstances are particularly extenuating, you need to clear the incomplete within 30 days of the final exam.

Honor Code: Although you are encouraged to discuss ideas with others, your programs are to be completed independently and should be original work. Whenever you obtain significant help (from other students, the section leaders, students in other classes) you should acknowledge this in your program write-up, e.g. "The idea to use radix sort to alphabetize the list of names was actually my section leader's idea." Any assistance that is not given proper citation will be considered a violation of the Stanford Honor Code. The Honor Code is taken very seriously in this class; any problems will be swiftly referred to the Office of Judicial Affairs.

To be even more specific, you are not allowed to collaborate on the coding of your programs, nor are you allowed to copy programs or parts of programs from other students. The following three activities are considered Honor Code violations in this course:

1. Looking at another student's code.
2. Showing another student your code.
3. Discussing assignments in such detail that you duplicate a portion of someone else's code in your own program.

Unfortunately, the Computer Science Department sees much, much more than its share of plagiarism. Because it's important that all cases of academic dishonesty are identified for the sake of those playing by the rules, we reserve our right to use software tools to compare your submissions against those of all other current and past CS106 students. It isn't our intent to create a Big Brother environment with moles and surveillance cameras. We're not creating that at all. We're just being clear about how far we'll go to make sure the consistently honest feel their honesty is valued and rewarded. If the thought of copying code has never crossed your mind, then you needn't worry, because I've never witnessed a false accusation. But if you're ever tempted to share code—whether it's because you don't understand the material or you do understand but just don't have enough time to get the work done—then you need to remember this paragraph is here.