CS106X: Programming Abstractions in C++
Course Information and Syllabus
Stanford University - Autumn 2015

Instructor

name: Cynthia Lee
email: cbl@stanford edu
office: Gates 191

TA

name: Ilan Goodman
email: igoodman@stanford.edu
office: TBD

Course Website

Main site: http://cs106x.stanford.edu/
Q&A forum: https://piazza.com/class/iekocotnibv144 (self-register)

Course Overview

Students coming to CS106X are expected to have substantial prior experience with programming constructs such as loops, functions/methods, arrays, console and file I/O, standard data types, and classes. In other words, students coming to CS106X are expected to know how to write code. In this course, we transform you from someone who knows how to write some code into someone who can wield the power of the computer in a more expansive, versatile, and fluent way. This course adds powerful new tools to your skillset: managing complexity through abstraction, using recursion to crack big, tough problems by breaking them into smaller solvable problems, and applying analytical tools to evaluate the efficiency of different software design choices.

Welcome to the next level, and welcome to CS106X!

Units

Undergraduates must take the course for 5 units. Graduate and professional students may opt for 3-5 units. Course requirements and grading are the same regardless of number of units.

Prerequisites

Substantial prior programming experience can take the form of strong performance in Stanford's CS106A or a rigorous high school AP CS course or equivalent. No extracurricular coding experience is necessary or expected, though students may be successful in 106X with that background and *without* having prior formal coursework. This course is taught in C++, but no prior exposure to C++ is expected. Most students come into 106X knowing Java, Python, or C++ (in roughly descending order of frequency). Students whose prior experience is in non-object-oriented or scripting languages may find that CS106A or B is better suited to their needs.

B or X?

Many students inquire about differences between CS106B and CS106X. It is important to know that ***B and X cover the same topics***.[1] CS106X covers the topics at a slightly accelerated pace, in order to accommodate one additional code project at the end of the quarter. What follows are remarks from my unofficial take on what distinguishes the two courses. You may also wish to consult with the B instructor, or even attend both classes for a week.

[1] Many years ago, X had both CS106A and B material packed into one quarter. This is no longer the case. I mention it because of rumors that persist despite this having changed many years ago.

Throughout the quarter, X assignments are characterized by greater difficulty in complexity of required features and independent design decision-making. Although B and X cover the same topics, B treats some topics on a more practical/coding level, compared to a more design-oriented, mathematical, or theoretical flavor in X. For example, where B lectures might cover several examples of declaring functions, X lectures might rush or skip the basic examples and move on to exploring why C++'s function declarations enable its compiler to be more efficient. Topics we treat with more math and theory include algorithm performance analysis and graph structures. Some of what we do anticipates future CS courses; thus **if you are simply interested in adding coding to your skillset, but not majoring in CS, B might be a more efficient path to your goal**. For instance, I would recommend to most graduate students from other departments that they take B.

- **Extensive further advice on choosing between A, B, X, and selected other CS courses is available from the CS106 course placement document: http://stanford.edu/~cbl/SSEA/Course%20Placement.pdf**

Topics

The following is our *approximate* course topics schedule for the quarter (subject to change). The terms on this list might not entirely make sense to you until we begin to cover them, but I list them here so that those who might be candidates for skipping CS106B/X entirely, or for whom other classes might more directly address their education goals, can make that determination:

- Week 1: Transition to C++ (very quick, anticipating that most of the audience will have used Java and not C++); Get started on the Basic ADTs (see week 2)
- Week 2: Basic ADTs: Vector, Grid, Stack, Queue, Map, Set, Lexicon.
- Week 3: Recursion, advanced recursion, backtracking recursion
- Week 4: "Big O" mathematical performance analysis; Classes in C++
- Week 5: Pointers and dynamic memory allocation (new/delete); Linked lists
- Weeks 6-7: Basic ADTs revisited: how do we implement them behind the scenes? Linked lists, trees, Binary Search Trees, tries, hashing
- Week 8: Graphs, graph search algorithms (BFS, DFS, A*), minimum spanning tree algorithms
- Week 9: Inheritance and polymorphism in C++
- Week 10: Special topics, quarter wrap-up

Textbook

Roberts, Eric. *Programming Abstractions in C++*. ISBN 978-0133454840. Required. There is also a PDF version available for free: http://cs.stanford.edu/people/eroberts//CS106BX-Reader.pdf

Class Environment

CS106X is an extremely difficult course, by design. We want to effect the greatest possible increase in your skills and performance as a programmer in the short 10 weeks that we have together. To make that happen for everyone, I strive to create an inclusive and equitable classroom, and I further depend on you to help each other obtain excellence rather than mistaking Stanford and this class for some kind of zero-sum game. **Please do your part by seeking to promote the success of others, and by treating each other in ways that respect and celebrate the diversity of talent that is drawn to our exciting and dynamic field of Computer Science.**

Here are a few specific things you should know about our policies on creating an inclusive and equitable class:

- **Office of Accessible Education accommodations:** If you have an OAE letter, please present it to me at your earliest convenience, so I can ensure that the course materials and staff support comply with your needs. I am always willing to do what it takes to support you, but I ask that you have your exam scheduling requests submitted no later than 1 week prior to the exam, because I must respect our room scheduling staff person's time to make those arrangements.

- **Preferred name and preferred pronouns:** I want you to be you in our class. You are always welcome to write your preferred name on all class assignments and exams. If you have a name and/or pronoun preference that doesn't match what our class roster gets from the registrar's office, please let me know and I will ensure we use that in our class.
- **Class expenses (textbook, device for class participation, etc.):** If obtaining any material for use in our class presents a financial hardship for you, please let me know and I will be happy to provide gift or loan items for you as needed.
- **Feedback:** Please do not hesitate to reach out to me, anonymously if you prefer, if any aspect of our course or class community can be improved.

Grades

Your overall course grade will be determined by **your choice** of the following two percentage breakdowns:

With lecture participation credit		*Without lecture participation credit*	
5%	section participation	5%	section participation
5%	**lecture participation**		
25%	midterm exam	25%	midterm exam
25%	final exam	**30%**	**final exam**
40%	homework assignments	40%	homework assignments

You are not required to formally notify me of your grading criteria choice; my spreadsheet that calculates final grades will simply apply the formula most favorable to you. Grades will be assigned *no stricter than* the usual scale of 90% = A- or better, 80% = B- or better, etc. A more generous curve may be adopted at the instructor's discretion, depending on exam difficulty or other factors.

Section Participation

You must sign up for a weekly 50-minute section, held on various times/places on Wed-Fri. **The section signup process will begin at the end of the first week of the quarter (details will be sent by email).** After the signup period, a matching process is undertaken by the course staff, and your section assignments will be e-mailed out to you. Sections begin the **second** week of classes. Although Axess lists sections for this course, **we don't look at Axess when assigning sections**. If you miss the section signup period, there will be a second round chance for you to enroll in a section, but you will be limited to sections that are not full; if this happens to you, just wait and watch for an email with those instructions.

Your section leader ("SL") will be your first point of contact for questions about homework or other course topics, grade your homework, provide individual guidance to you throughout the quarter. In section, your SL discuss sample problems in more detail than we can in lecture. Part of your course grade comes from attending and participating in your section and in code review meetings ("interactive grading" or "IGs") on a regular basis.

Lecture Participation

There will be frequent opportunities for participation during lecture: solving problems, responding with mobile app, and group discussion. **Lecture attendance is not required**, but **everyone present on a given day is expected to sit in groups of about three students and contribute to their group's discussions**.

This class operates more like a conversation than other college classes you may be used to.

Discussion questions with mobile app votes are used to help me gauge class understanding in real time (rather than waiting until the midterm) and tailor lecture to your needs. To recognize the effort these votes require on your part, I offer optional credit for regular class participation. If you attend and participate actively in lecture (phone app *and* active group discussion), this can count to your final grade. You may still earn 100% of this credit with **up to six absences**. In order to accommodate illnesses, vacation, sports competitions, family events, emergencies, etc., but still recognize your otherwise consistent efforts to participate, 6 days are automatically

dropped from your score (there is no extra credit for <6 absences, absences in excess of 6 are scored as *days_attended*/(*total_days*-6)). **There is no need to email the instructor about the reason for your absence**—it is automatically counted.

Exams

Midterm exam:

- Tuesday, October 27th, 7-9 p.m., *Location TBA*

Final exam:

- Monday, December 7th, 8:30-11:30 a.m., *Location TBA*
- The final exam is comprehensive (covers topics from the entire quarter), if for no other reason than it would be impossible not to cover topics from the first half of the quarter because they are so fundamental to the subsequent material.

If you must miss an exam due to an unavoidable academic schedule conflict (an exam in another class at the same time), **you must contact the instructor immediately**. **There are no discretionary alternate exams. Vacation, internship, etc., are not allowable reasons to miss an exam.**

106X exams are **open-book (the official course textbook), but closed-notes/handouts except one page of notes**. I know textbooks are expensive and that some students choose not to purchase one, so a few copies will be available for loan during the exam. A reference page will be included in the exam itself, containing commonly needed information (for example, documentation on the Stanford C++ library classes). The purpose of my open-book, closed-notes policy is to reduce the need for rote memorization of anything (this is a problem-solving course, not a memorization course), but still standardize the playing field in terms of precisely what information students have with them.

Homework Assignments

Homework consists of programming assignments done on an individual or pair basis. Pair programming is optional though encouraged on the assignments where it is allowed (some assignments, like the first assignment, are strictly individual). If you work on an assignment as a pair, you must submit as a pair, and you will be given the same grade.

You will have roughly 7-10 days to work on each assignment, and there will be 7 assignments in total (with some time off around the midterm).

Programs will be graded on both functionality and style. Rather than a direct point-based scoring system, a check-plus, check, check-minus grading system is used. Dividing the grades into these general categories means that your section leader can spend more of the interactive grading ("IG") time talking about what you need to learn from the assignment without being bogged down in accounting for minor point deductions. For each assignment, you must make an appointment for an IG session with your section leader. Your section leader will explain in section how to schedule these sessions and go over the grading process and standards in more detail.

From past experience, most grades will be ✓ or ✓+. The meanings of the grades are as follows, with *approximate* letter grade equivalents:

0	**Zero:** No credit. Typically this is only given if you do not submit the assignment or it shows no effort whatsoever. *(F grade)*
– –	**Minus-Minus:** Has extremely serious issues; shows almost no understanding *(D or F grade)*
–	**Minus:** Has very serious issues and does not demonstrate significant effort and understanding. *(C or D grade)*
✓–	**Check-Minus:** Has problems serious enough to fall short of the expectations/requirements for the assignment. *(B- grade)*
✓	**Check:** Meets the requirements for the assignment, with a few small problems or areas of improvement. *(B or B+ grade)*

✓+ **Check-Plus:** Satisfies all the requirements for the assignment, showing very solid functionality as well as good style. It reflects a job well done. It is rare to receive a mark above +, so this is generally the score to aim for. *(A- or A grade)*

+, ++ **Plus, Plus-Plus:** Excellent; significantly exceeds our expectations for the assignment. To receive these grades, a program often reflects additional work beyond the requirements and/or solves the problem in a particularly elegant way. A mark of + or ++ will be given out only on rare occasions and should not be considered a standard goal for each assignment you submit. It is more of a way of congratulating the rare submission that goes above and beyond. *(A+ grade)*

An assignment's style score may be capped at a given maximum if its functionality score is particularly low.

Working in Pairs

Some the assignments must be completed on an individual basis, but the majority of them allow you to optionally work in a pair with a partner. Working in pairs can improve student learning by giving you someone to talk to when they are stuck, or by letting you see a different way of approaching the same problem. Submitting work in pairs also reduces our staff grading workload which is important given our current rapidly increasing enrollments in CS.

If you choose to work with a partner, you must pair with another student who is currently taking the course and 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 working in a pair must be taking the course for the same grading basis. That is, a student taking the course Credit / No Credit may not pair with one taking it for a letter grade. 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 enrolled in the course may be part of any pair.

If you submit 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. Regardless of pairs, every student is still responsible for learning all course material. Of course, all exams are completed individually.

Many details about working in pairs are answered in a FAQ linked to from the class web site assignments page. Please read that page and make sure that you follow its guidelines.

Assignment Submission and Lateness

Submit your assignments electronically through a link provided on the course web site. Each student begins the course with **3 "late days"** for use on homework assignments. A late day allows you to submit a program **up to one lecture day later without penalty**. For example, if a program is due on Monday at 1pm, using one late day allows you to submit it on Wedesnday at noon without penalty. If a program is due on Friday at 1pm, using a late day allows you to submit it on Monday at noon without penalty. **You may use up to 2 late days on any given assignment; you may not use all 3 on the same program.** Any fraction of a late day counts as one day. For example, if a program is due on Monday at 1pm, turning in the assignment at 5pm will consume an entire late day. If you turn in an assignment late and do not have any more late days, you will lose one bucket grade per 24 hours.

In no case may any assignment be turned in more than 3 lecture days late. Graders have deadlines for completing their work and it is not fair to them to delay their work more than this, so the assignment will be given zero.

Late days **may not be used on the very last assignment**, even if you have late days remaining, and you may not turn in an assignment past this deadline for a deduction.

Late days are intended to be used for emergencies that would typically warrant exceptions granted by the instructor: illness, accidents, family emergencies, and the like. Please use care in deciding when to use your late

days, because students who have consumed late days for reasons other than dire emergency will **not** be granted additional extensions, **even in cases of dire emergency**.

Help Resources

The department operates a student computer cluster/lab called the "**LaIR**" located on the second floor of Tressider Union. Section leaders will be available at the LaIR to help students with problems. Please refer to the course website for the most current LaIR schedule information.

Students are also invited to post questions (but **no source code**) to the Piazza web discussion board for our class. Emailing your SL is another great idea.

Who Should I Email/Contact?

- **Miscellaneous basic policy questions** (When is the midterm? How many late days do I get?): Re-read this syllabus/course information document, any additional questions post to Piazza (unless sensitive or confidential)
- **Help with assignments or course topics:** Piazza or SL
- **Assignment special circumstances/requests for additional late days:** TA
- **Anything very sensitive or confidential:** TA or Cynthia Lee
- Career or course advising, questions about research and other opportunities at Stanford, concerns about performance in the class, suggestions for improving the class, or anything that really only the instructor can answer: Cynthia Lee