CS106B
Winter 2021

Handout 00
January 11, 2021

Course Information

Course Overview

A nonprofit needs to assign tasks to its volunteers so they get completed as fast as possible. A sociologist wants to learn how Americans utilize their National Parks system. A climatologist wants to estimate how many trees are in a forest. A city planner wants to see whether a proposed development will be underwater as sea levels rise. A physicist wants to know why perturbations to a system cause cascading effects.

How can we use computing power to answer these questions?

This course is about transitioning from “I know how to write programs” to “I know how to solve problems with computers.” Over the course of ten weeks, we’ll explore an array of techniques, tools, and perspectives useful for modeling and solving problems. We’ll explore recursion and see how it can be used both to model the intricacies of nature and to optimally allocate resources. We’ll develop a rich vocabulary of structures that capture both the position of a dancer in space and ways of scheduling patients in a hospital. And we’ll see how to put these techniques to use in problems drawn from a range of disciplines. By the time you’ve completed this course, you’ll learn how to look at problems in fundamentally different ways and how to use those perspectives to create clean and elegant computational solutions.

Instructor

Keith Schwarz (htiek@cs.stanford.edu)
Office Hours: Tuesdays, 10:00AM – 12:00PM Pacific time.
Call into my office hours to chat about the class, the assignments, life at Stanford, etc.

“Chat About Whatever” Hours: Fridays, 3:00PM – 5:00PM Pacific time.
Call into my Chat About Whatever hours to talk about whatever is on your mind!

Head TA

Chase Davis (cs106b-ta@cs.stanford.edu)
Office Hours: Mondays, 8:00AM – 10:00AM and Wednesdays, 8:00AM – 10:00PM.

(All times are Pacific time.)

Website

The class website is https://cs106b.stanford.edu and it's loaded with resources for this course. There, you'll find all the handouts and lecture materials, along with additional links you may find useful. I would suggest periodically polling the website to stay on top of any important developments in the course.

Lectures

Lectures will be offered live on Mondays, Wednesdays, and Fridays from 11:30AM – 12:30PM over Zoom. If you can’t make that time slot, no worries! We’ll post a recording of the lecture up on Canvas.

Discussion Sections

In addition to lecture, you must also sign up for a weekly discussion section. Section signups are handled online and run from Thursday, January 14 at 5:00PM to Sunday, January 17 at 5:00PM (Pacific time) at http://cs198.stanford.edu/section. After a matching process, your section assignment will be emailed by Tuesday, January 19th. Sections begin the second week of classes. This link is also available on the CS106B web page.

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.

Getting Help

Learning to program is like learning any skill – it takes practice, and it's normal to need some help from time to time.

You can get (virtual) in-person help from our staff Sundays through Thursdays through "LaIR Hours." Use this link to sign up:

https://cs198.stanford.edu/lair/student

LaIR hours run Sundays through Thursdays from 5:00PM – 9:00PM Pacific time. Log in using the link above and join the LaIR queue. You'll then be contacted over Zoom by the next free section leader for a one-on-one help session.

If you can't make the 5:00PM – 9:00PM time slot, we also hold LaIR hours on Tuesday and Thursday from 9:00AM – 11:00AM. ***These time slots are reserved only for students who cannot make the normal LaIR time slots*** (for example, folks calling in from the UK who would prefer to get some sleep), so please don't sign in during this time if you can comfortably make the standard times.

In addition to the LaIR, you can get help asynchronously on our online EdStem forum. This is a Q&A site that we're constantly monitoring, and we're happy to take conceptual questions there. Visit Canvas to get a link to the EdStem page. Our EdStem form is primarily for conceptual questions. If you have a coding question, please post it privately so you don't share your code with everyone else in the class.

And finally, you're welcome to call into Chase or Keith's office hours each week. We're looking forward to meeting you and are happy to help out!

Units

If you are an undergraduate or a non-matriculated graduate student, you need to enroll in CS106B for five units (these are department and university policies, respectively). If you are a matriculated graduate student, you may enroll for anywhere between three and 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. The unit flexibility is simply to make scheduling easier for matriculated graduate students.

Five-unit courses at Stanford vary greatly in their difficulty. Based on past student experiences, you should expect that this course probably will require a time investment proportional to its unit load. Expect to put in around 10 – 15 hours each week working on CS106B. We'll offer a lot of support through office hours, extra practice problems, and practice exams, and if you're willing to put in the effort to learn the material, the course staff will be behind you every step of the way.

Prerequisites

CS106B assumes that you have a familiarity with programming at the level of CS106A or the AP Computer Science exam. 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.

A note: although this class uses C++, this class is primarily designed to teach abstraction, recursion, and algorithmic analysis. If you already know those topics and just want to learn C++, you may want to opt to take CS106L instead of CS106B.

Readings

The required reading for this course is Eric Roberts' *Programming Abstractions in C++*. You can purchase a copy at the bookstore using this link:

https://www.bkstr.com/stanfordstore/search/keyword/programming%2520abstractions

You can also use this online version, which is a 2012 draft version of the book. It isn't as polished as the final one, but should be sufficient for the quarter:

http://web.stanford.edu/dept/cs_edu/BXReader-Beta-2012.pdf

We assume that the majority of you have no prior programming experience in C++, and this textbook is a great resource to use at the start of the quarter as you're transitioning into the language.

Assignments

There are nine programming assignments this quarter. Except for the last assignment, each assignment is graded in a one-on-one session with your section leader, who rates it according to the following scale on functionality and style:

`++` 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 head TA.

`+` 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 understanding.

`0` A submission that was either not submitted or 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.

For each assignment (except the last), 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. Typically, functionality scores are weighted at two-thirds of the total score for an assignment, while style scores are weighted at one-third of the total score.

Working in Pairs

A few of the assignments in this course must be completed on an individual basis, but the majority 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 the submission.

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 11:30AM, Pacific time, on their specified due dates. If you submit by this time, we’ll award you a small “on time” bonus that will slightly increase your overall raw score at the end of the quarter.

There is a 48-hour grace period after this deadline during which you can submit the assignment for full credit, though we won’t award the “on time” bonus. You can think of this 48-hour grace period as a way of automatically granting yourself an extension on any assignment in which you need it. After this 48-hour grace period has expired, no late submissions will be accepted.

If you have a medical or family emergency and will need an extension beyond 48 hours, please contact the head TA, Chase Davis (cs106b-ta@cs.stanford.edu), no more than 24 hours before the normal assignment deadline. ***Only Chase 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.

Similarly, if you have OAE accommodations and will need an extension, please feel free to contact Chase.

Midterms

In addition to the programming assignments, there will be two midterm exams. The first goes out Friday, February 12th and is due on Sunday, February 14th. The second goes out Friday, March 12th and is due Sunday, March 14th. We will release more information about the midterm exams once we get a bit closer to the exam dates.

Grading

Overall, your grade for this course will be determined as

Programming Assignments:	**45%**
First Midterm Exam:	**25%**
Second Midterm Exam:	**25%**
Section Participation:	**5%**

We assign letter grades as follows. We first determine a grading curve over raw scores to assign initial grades. We never assign grades that are lower than the decile of your raw score; for example, a 90% will never map to anything lower than an A-. Assuming that both your assignment score and your composite exam score represent passing work, you will receive the letter grade assigned by the grading curve. Otherwise, you will receive a non-passing grade. (The numbers denote “passing work” are set at the discretion of the instructor. We will likely use 60% as a cutoff for passing work for programming assignments and 50% as a cutoff for passing work for exams, subject to change.) Your final grade will be determined solely as mentioned above. We do not offer any make-up work.

Incomplete Policy

If a serious medical or family emergency arises and you cannot complete the work in this course, you may contact Keith – not the head TA and not the section leaders – to request an incomplete. We reserve incompletes only for emergencies, so we do not grant incomplete grades for poor performance on the assignments or exams, nor do we offer incompletes for busy work schedules.

To be eligible for an incomplete, you must have completed all of the assignments (except possibly the most-recently-due assignment) and have a solid academic performance in the course, as determined by the instructor. The instructor has final say in whether to grant or deny incompletes. The above criteria indicate certain cases in which incompletes will not be granted, but there are no situations where the instructor is obligated to offer an incomplete.