

## 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](mailto:htiek@cs.stanford.edu))  
Office Hours: Tuesdays, 10:00AM – 12:00PM, Gates 172.
- Head TA** Katherine Erdman ([kerdman@stanford.edu](mailto:kerdman@stanford.edu))  
Office Hours: Tuesdays and Thursdays, 3:00PM – 4:15PM, Gates B02.
- 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** Mondays, Wednesdays, and Fridays from 11:30AM – 12:20PM in Hewlett 200. Lectures are not recorded; you are encouraged to attend lectures and ask questions.
- Discussion Sections** In addition to lecture, you must also sign up for a weekly 50-minute section. Section signups are handled online and run from Thursday, January 9 at 5:00PM to Sunday, January 12 at 5:00PM at <http://cs198.stanford.edu/section>. After a matching process, your section assignment will be emailed by Tuesday, January 14th. Sections begin the second week of classes. This link is also available on the CS106B web page.
- Although Axxess lists discussion sections for this course, we don't look at Axxess section enrollments when assigning sections. Even if you're enrolled in a section through Axxess, you need to sign up through our system as well to make sure that someone will be grading your assignments.
- 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 have any questions on the material or assignments, feel free to stop by the the LaIR or CLaIR between 7PM and 11PM, 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 schedule at <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 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.

## CS106L

CS106L is an optional, one-unit companion course to CS106B that dives deeper into the beautiful world of the C++ programming language. 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.

## CS106S

CS106S is an optional, one-unit companion 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' *Programming Abstractions in C++*. It's available at the bookstore and there are copies on reserve in the engineering library.

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 eight 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, 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 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 periods. A “late period” is a class period (i.e. Monday to Wednesday counts as one late period). ***No assignments may be submitted more than two class periods late.*** If you submit an assignment late and have used your late periods, we will cap your score on both functionality and style at ✓- for the purposes of grading (though in your interactive grading session you’ll still receive the scores you would have otherwise received on the assignment so that you get more detailed feedback). You should think of these free “late periods” 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 periods” will generally not be granted. In very special circumstances (primarily extended medical problems or other emergencies), extensions may be granted beyond the late periods. All extension requests must be directed to the head TA, Katherine Erdman, no later than 24 hours before the program is due. ***Only Katherine 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 periods 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 periods, since university policy prevents us from having assignments come due then.

## Grading

In addition to the programming assignments, there will be a midterm exam and a final exam. The midterm will be held on Tuesday, February 11<sup>th</sup> from 7:00PM – 10:00PM, location TBA. The final exam will be held on Monday, March 16<sup>th</sup> 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

<b>Programming Assignments:</b>	<b>35%</b>
<b>Midterm Exam:</b>	<b>25%</b>
<b>Final Exam:</b>	<b>35%</b>
<b>Section Participation:</b>	<b>5%</b>

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 (with the two exams weighted above) 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.