

Course Information

- Course Overview** Welcome to CS106A! CS106A is a first course in computer programming and software engineering. In this course, you'll learn the different techniques, programming constructs, and design strategies that form the basis for modern software. By the time you've completed the course, you'll be comfortable writing computer programs that interact with the user, process and manipulate data, and report information back textually and graphically. More importantly, though, you'll learn how to approach problem solving from a computational perspective and will gain an exposure to different areas within computer science and how programming is applicable across all sorts of domains.
- Instructor** Keith Schwarz (htiek@cs.stanford.edu)
Office: Gates 178
Office Phone: (650) 723-4350
Office Hours: TBA
- Head TA** Vikas Yendluri (vikasuy@cs.stanford.edu)
Office: Gates 160
Office Hours: Tuesday/Thursday, 2:15PM – 4:15PM
- Website** The course website is <http://cs106a.stanford.edu> and it's loaded with useful resources. There, you'll find all the handouts for this course, lecture slides, lecture code, assignment starter code, software links, relevant news articles, and much more. I would suggest periodically polling the website to stay abreast of important developments in the course.
- Lectures** This quarter, we're offering two lectures of CS106A each day. The lectures should be pretty much identical and you're welcome to attend whichever of the two lectures you'd like. One lecture will be on Monday, Wednesday, and Friday from 10:00AM – 10:50AM in room 420-040 and the other will be Monday, Wednesday, and Friday from 3:15PM – 4:05PM in Nvidia Auditorium in the Huang Engineering Center. The 3:15PM lecture will be recorded and posted online through SCPD, and there will be a link to the videos on the course website.
- Prerequisites** There are no prerequisites for this course, and everyone is welcome!
If you already have some familiarity with programming, CS106A may still be an appropriate course for you, but you may want to consider moving on to one of the more advanced programming courses. Handout #02 contains placement information that you may find useful in deciding whether CS106A is right for you. As always, feel free to get in touch with us if you have any questions.
- Units** If you are an undergraduate, you must enroll for five units. If you are a graduate student, you may enroll for between three and five units, depending on what best fits into your schedule. If you're taking the course through SCPD, you are required to enroll for five units. Regardless of how many units you are enrolled for, the course content and requirements will be the same.

Discussion Sessions In addition to lecture, you must also sign up for a weekly 50-minute section. Section signups are handled online at <http://cs198.stanford.edu/section>. Section signups open on Thursday, January 9 at 5:00PM and close Sunday, January 12 at 5:00PM. After a matching process, we will send you an email with section assignments by Tuesday, January 14. Sections begin the second week of classes. This link is also available on the CS106A web page. Although Axxess lists discussion sections for this course, it's not necessary to enroll in a discussion section on Axxess. We'll handle everything through the CS198 website mentioned above.

Section Leaders The team of CS106A Section Leaders and Course Helpers are here to help make this course the best that it can be. If you need help on any of the assignments, feel free to stop by the Tresidder LaIR between 6PM and midnight, Sundays through Thursdays. You can view the most current helper schedule by going to <http://cs198.stanford.edu> and clicking the "Helper Schedule" link. This link is available on the CS106A web page.

Readings This course has two required readings:

Karel the Robot Learns Java. This 35-page tutorial introduces major concepts in programming through Karel the Robot, a friendly robot who runs around in a grid world. In the first week of the quarter, we'll use Karel to explore programming, problem solving, and software engineering. This course reader will be an invaluable resource as you work through the first assignment. The reader is available electronically on the course website and in hardcopy from the Stanford Bookstore.

The Art and Science of Java by Eric Roberts. Once we've acclimated to the wonderful world of Karel, we'll begin building larger and more elaborate programs using the Java programming language. *The Art and Science of Java* is a fantastic introduction to software engineering in Java, and you will definitely want to have a copy as we start moving into more interesting and advanced topics.

In addition to these readings, we'll be periodically distributing handouts in class. These handouts, which will also be available online at the course website, should help supplement the treatment of the material given in the other readings. We will store extra copies of the handouts in the "handout hangout" (the B wing of the first floor of the Gates building), so don't worry if you forget to pick up a copy.

Computers As in any programming course, the assignments in CS106A require extensive hands-on use of a computer. The preferred platform for doing the work is the Eclipse development environment, which runs under both Mac OS X and Microsoft Windows. (It is also possible to use run these programs in Linux; talk to Keith or Vikas about getting that set up). Instructions on obtaining and using the Eclipse environment – which is free to download – will be distributed in a separate handout.

Assignments There will be **seven** programming 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 slightly more difficult and require more time as the quarter progresses, so the later assignments are weighed slightly more than the earlier ones. Except for Assignment #7 (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:

- ++ An absolutely fantastic submission of the sort that will only come along a few times during the quarter. Any grade of ++ must be approved by the instructor and TA.
- + A submission that is "perfect" or exceeds our standard expectation 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.

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. The interactive-grading session with your section leader must be scheduled within two weeks of the due date.

Late policy

Each of the assignments is due at 3:15P.M. 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 3:15P.M. sharp on the dates indicated on the assignment handout. Anything that comes in after 3:15P.M. will be considered late.

Everyone gets a bit overwhelmed every now and then, so to give you some flexibility with assignment due dates, everyone begins the quarter with **two** free late periods. Each late period lets you submit one assignment one class period late. For example, an assignment due on Monday could be submitted on Wednesday using a late period, and an assignment due on Friday could be submitted on Monday using a late period. Late periods do factor in national holidays, so an assignment due on the Friday before MLK day or Presidents Day could be submitted on the following Wednesday using one late period. You may use both late days on a single assignment if you'd like, though we strongly advise against this as you'll almost certainly fall behind in the course.

If you submit an assignment late after using all your late periods, we will assign a 1% penalty to your overall grade in the course per class period late. No submissions will be accepted for any assignment more than three class periods after the stated due date, even if you use late periods.

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 (medical or family emergencies, religious reasons, etc.), extensions may be granted. All extension requests must be directed to the head TA, Vikas Yendluri, no later than 24 hours before the program is due. **Only Vikas can approve extensions.** In particular, please do not ask your section leader for an extension, since they cannot grant you one.

Note that late days may not be used on Assignment 7, because it is due during the time slot normally reserved for the CS106A final exam.

Grading

In addition to the seven programming assignments, there will be two midterm exams. The first midterm exam will be held on **Wednesday, February 12** from **7PM – 10PM**, and the second will be held on **Wednesday, March 5** from **7PM – 10PM**. If you have a conflict that will prevent you from taking either of the exams, we'd be happy to try to find an alternate time. However, you must let us know about this no later than **one week in advance** so that we have adequate time to schedule another exam room and time. Note that no alternate exams will be given after the scheduled exam times, so all alternate exams must be taken prior to the standard exam time.

Overall, your grade for this course will be determined as

Programming Assignments:	55%
First Midterm:	20%
Second Midterm:	20%
Section Participation:	5%

Note that we do not offer “make-up” work that you can use to increase your grade if you perform poorly on the problem sets or exams, though by adding extensions to the assignments we will award extra credit that can be used to raise your overall grade.

Incomplete Policy

If you have a serious medical or family emergency and cannot complete the work in this course, you may contact Keith to arrange for an incomplete. Except for medical or family emergencies that arise in the last week of the quarter, no requests for incompletes will be considered during or after the final week of class. Note that we do not grant incomplete grades for poor performance on the assignments, nor do we grant incomplete grades for busy work schedules.

In order to receive an incomplete in the course, you must have completed all assignments in the course at the time at which you request an incomplete, except for possibly the most-recently-due assignment.