# Course Information

| | |
|---|---|
| **Instructor** | Keith Schwarz (htiek@cs.stanford.edu)<br>Office: Gates 178<br>Office Phone: (650) 723-4350<br>Office Hours: Tuesday, 2:00PM – 4:00PM |
| **Head TA** | Gil Shotan (gilsho@stanford.edu)<br>Office: Gates 160<br>Office Hours: Monday/Wednesday, 10:00AM – 11:59AM |
| **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** | Mondays, Wednesdays, and Fridays, 3:15PM – 4:05PM in NVIDIA Auditorium. |
| **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 10 at 5:00PM and close Sunday, January 13 at 5:00PM. After a matching process, we will send you an email with section assignments by Tuesday, January 15. Sections begin the second week of classes (i.e., next week). This link is also available on the CS106A web page.<br><br>Although Axess lists discussion sections for this course, you don't need to (and shouldn't) enroll in a discussion section through Axess. 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. For an up-to-date schedule of helper hours, you can view the most current helper schedule by going to http://cs198.stanford.edu and clicking the "Helper Schedule" link. Again, this link is available on the CS106A web page. |
| **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. Regardless of how many units you are enrolled for, the course content and requirements will be the same. |
| **Prerequisites** | There are no prerequisites for this course, and everyone is welcome!<br><br>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. |

| | |
|---|---|
| **Email** | We will use email as a primary means for distributing announcements and notifying you when your assignments has been graded, so having an email account is a requirement for this course. If you don't already have an email account, you can get one through LaIR.

Information on obtaining an account is available at the Tresidder computer cluster, from your Resident Computer Consultant, or via the web at http://email.stanford.edu. |
| **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 Gil about getting that set up). Instructions on obtaining and using the Eclipse environment – which is an open-source software project and free to download – will be distributed in a separate class 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. To ensure that this score is given only rarely, any grade of ++ must be approved by the instructor and TA. Since your section leader would almost certainly want to show off any assignment worthy of a ++, this review process should be straightforward.

+ 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.

**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 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 days." "Late days" are **class days**, not actual days (i.e. Monday to Wednesday counts as one late day, as does Friday to Monday). After late days are exhausted, programs that come in late (up to a maximum of three class days) will be assessed a late penalty of one grade bucket per day (e.g., a ✓+ turns into a ✓, and so forth). 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.

Assignments received later than three class days following the due date will **not** be graded, so please make an effort to get the assignments completed on-time. The interactive-grading session with your section leader must be scheduled within two weeks of the due date.

You should think of these free "late days" 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 days" will generally not be granted. In very special circumstances (medical or family emergencies, religious reasons, etc.), extensions may be granted beyond the late days. All extension requests must be directed to the head TA, Gil Shotan, no later than 24 hours before the program is due. **Only Gil will be able to approve extensions**. In particular, please do not ask your section leader for an extension, since they cannot grant you one.

**Grading**

In addition to the seven programming assignments, there will be two midterm exams. The first midterm exam will be held on **Monday, February 11** from **7:00PM – 10:00PM**, and the second midterm exam will be held on **Monday, March 11** from **7:00PM – 10:00PM**, location TBA. 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 **two weeks in advance** so that we have adequate time to schedule another exam room and time.

Overall, your grade for this course will be determined as

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