Chris Piech
CS 106A
Handout #1
Apr 2nd, 2018

CS 106A — General Information

Professor: Chris Piech

E-mail: `piech@cs.stanford.edu`
Office: Gates 193
Office hours: Thursday 1:30P.M.–3:30P.M.

Head TA: Julia Daniel

E-mail: `jdaniel7@stanford.edu`
Office: Gates B02 (in the basement)
Office hours: TBD.

Class web page

The web page for CS 106A is located at **`http://cs106a.stanford.edu`**

You should regularly check the class web site for handouts, announcements and other information, including the most up-to-the-date information on assignments and errata. **Please note that the class web page will have links to essential class materials including electronic copies of class handouts and assignment files.**

Discussion sections

In addition to lecture, you must also sign up for a weekly 50-minute section. In order to take CS 106A, you must sign up for a section between 5:00P.M. Thursday, April 5th and 5:00P.M. Sunday, April 8th. Unlike Hamilton tickets, section sign ups are **not** first come first serve. The signup form will be available on the web at the URL **`http://cs198.stanford.edu/`**. After a matching process, your section assignments will be e-mailed out to you by Tuesday, April 10th. Sections begin the second week of classes (i.e., next week). Note that you should only sign up for sections at the URL indicated previously (you should not sign-up for sections on Axess).

As a side note, some of the programming assignments in this class will be done individually and for others you will have the option to work in pairs (more on that later). If you do decide to work in a pair for those assignments, you may only pair with someone in the same section as you. Please keep this in mind when signing up for sections.

Section leaders and course helpers

CS106A provides extensive assistance for students. Section Leaders and Course Helpers are available from Sunday through Thursday evenings each week in the Tresidder Union dining area. Sections Leaders in Tresidder are there to help you with your assignments (what we call Lair) or to help answer conceptual questions (what we call the Clair, for conceptual-lair). Check the web site **`http://cs198.stanford.edu/`** and click on the "Helper Schedule" link for the latest schedule of Helper Hours.

Units

If you are an undergraduate, you are required to take CS 106A for 5 units of credit. If you are a graduate student, you may enroll in CS 106A for 3 units if it is necessary for you to reduce your units for administrative reasons. Taking the course for reduced units does not imply any change in the course requirements.

Texts and handouts

There are two required texts for this class, both of which are available from the Stanford Bookstore. The first is a course reader entitled *Karel the Robot Learns Java*—a 35-page tutorial that introduces the major concepts in programming in the context of an extremely simple robot world. The second is the textbook *The Art and Science of Java* by Eric Roberts. In addition to these texts, we will also distribute additional material in the form of class handouts. Class handouts will be available electronically in PDF format on the CS 106A web site. If you prefer printed handouts, you can print a copy from the web.

Programming assignments

As you can see from the course schedule, there will be seven assignments (Assignment 1 – Assignment 7). The assignments will become slightly more difficult and require more time as the quarter progresses. Thus, the later assignments will be 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 not be too cumbersome.

+ 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 funtionality 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 over many quarters of experience. 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.

We have found, over many years of teaching CS106A, that *how* a student approaches a problem can be more indicative of their future programming success than whether or not they get the correct answer. In the spirit of always trying to offer a better CS106A, this quarter we are going to experiment with giving you feedback not just on your final submission, but also on the way you approach the problem. As you work through a problem, Eclipse automatically captures your intermediate work. When you submit a Stanford CS106A assignment, the intermediate work from that assignment is shared with your section leader. How you approach a problem over time (and your intermediate work) will *not* be used to determine your grade, only to help you learn!

Working in Pairs

Most of the assignments in this course must be completed on an individual basis, but some of them 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 student 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 pairings between assignments. In other words, you don't have to keep the same pairing for every assignment that allows pairs (and you can even choose to do some in pairs and others 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 person 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 help solve that program. **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 interative 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 ***11AM PST*** on the dates specified in the assignment handouts. The program code for your assignments must be submitted electronically as described in a separate handout. Anything that comes in after 11:00A.M. 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 days." "Late days" are class days, not actual days (i.e. from Monday to Wednesday is one late day). After the 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). Assignments received later than three class days following the due date will not be graded. The interactive-grading session with your section leader must be scheduled within two weeks of the due date. **Note that late days may not be used on the last assignment (#7) and no assignments will be accepted after the last day of classes (June 6th).**

If you are working in a pair and turn in an assignment late, both members of the pair will be assessed "late days". For example, if you turn in your assignment as a pair one day late, then both members of the pair each incur one "late day." If you are out of free "late days", but your partner isn't, then your assignment grade is penalized one grade "bucket", but your partner would simply use one of his/her free "late days" (and thus not be penalized one grade "bucket"). So you can think of "late days" being measured per student, and we apply any penalties *individually* for submissions that are made in pairs. Note: you cannot transfer free late days to your partner.

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, getting an extension beyond the two free "late days" will generally not be granted. In *very special* circumstances (primarily extended medical problems or other emergencies), extensions may be granted beyond the late days. All extension requests must be directed to the head TA no later than 24 hours before the program is due. Only the head TA will be able to approve extensions. In particular, do not ask your section leader.

Examinations

The midterm examination will be a two-hour test administered **outside of class from 7:00pm-9:00pm on Monday, May 7th**. If you have a conflict with this time, and absolutely cannot make the regularly scheduled midterm, you must send a request by electronic mail to me (**`piech@cs.stanford.edu`**) by 5:00pm on Monday, Apr 30th to arrange an alternate exam time. Any alternate midterm exam will be within at most one day (earlier or later) than the regular exam time, so make sure you are available in that time window if you cannot make the regular exam.

The final examination is scheduled for **Friday, June 8th from 8:30am-11:30am**. For a variety of reasons (including university policy), **there will be no alternate time for the**

final exam. Please make sure that you can attend the final exam at the specified time before enrolling in the class.

All examinations are open-book (class course reader and textbook only), and you may use any notes, handouts, or materials from the class, but you cannot use electronic devices of any type (i.e. portable computers, phones, etc).

Grading

Final grades for the course will be determined using the following weights:

45% Programming assignments (weighted toward the later assignments)
30% Final examination
15% Midterm examination
10% Section participation

Computer facilities

As in any programming course, the assignments in CS 106A 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 (XP, Vista, Windows 7 or Windows 8). Instructions on obtaining and using the Eclipse environment—which is an open-source software project and therefore free to download—will be distributed in on the course website.

That's all folks! Welcome to CS106A.