

CS 3 General Information

Website: <http://inst.eecs.berkeley.edu/~cs3>

Welcome to CS 3, "Introduction to Symbolic Programming". This course will introduce you to computer programming, using the Scheme programming language (a dialect of Lisp). We're going to do everything we can to make this summer SUPER fun and we hope you have a good time!

Am I prepared for this class? - YES

The only prerequisite to CS 3 is high school algebra: more specifically, familiarity with variables and simple functions. We don't assume that you have had any previous computing experience. If you have done some programming, especially involving the technique of recursion, you should seriously consider taking CS 61A instead of CS 3. Many people, however, find CS 3 a good precursor to CS 61A.

Why so much time in class?

You learn programming by doing it rather than by listening to us talk about it. Thus, CS 3 is organized to maximize your time designing and writing programs and experimenting with the programming environment. You'll work hard, but learn a lot. The lab sections and online activities are designed to help you get feedback at the time you need it; and we have staff in the lab to ensure that you're working productively rather than flailing around.

Staff

The instructor is Colleen Lewis (colleenL@berkeley.edu).

There is one TAs that will run the lab sessions during the week. The TA is Gilbert Chou.

There is also a reader who grades your homework and lab assistants who work with you during the lab sections. The reader is Ahmed Owainati and you can email him at cs3-ra@imail.eecs.berkeley.edu with any questions about homework grades.

Books and other course material

There is one required book for the course: *Simply Scheme*, by Brian Harvey and Matt Wright (second edition, MIT Press, 1999), is available at local bookstores. Some additional readings for the course are online. We will let you know if/when they can be purchased in printed form.

Class activities

Class time is Monday – Thursday 9-12 pm and Fridays 10-12. You should arrive to class on time, but you might be able to leave a bit early, depending upon the work load for that day.

For most of the summer, the typical lab period will involve a variety of activities, the majority provided online. It will start with a short lecture or activity followed by a quiz based on topics covered on the homework or in the preceding class. Each start-of-period quiz will count toward your course grade, and you have to take the majority of these in the lab classroom: we'll count up to 4 quizzes taken outside of lab. Following this, you'll be reading, experimenting, programming, brainstorming, debugging, evaluating each other's ideas, and working with partners. There will be three "mini-projects" during the semester, to which some of the lab meetings will be devoted. The last two weeks of the summer will be less structured, as you'll be working on a large programming project.

Lectures

Despite the posted schedule with lectures only on Friday, lectures will be held during the first 15-20 minutes of each class. These lectures will introduce and motivate new material or review confusion noticed during the previous lab sections. As a result, Friday will have the same format as the class does Monday – Thursday.

Surveys

In CS 3, you will be using programming tools and course material devised by a research group of computer science and education researchers. To determine the effectiveness of these tools and material, we are gathering data on your background and performance, via questionnaires, interviews, and analysis of your work. You will be expected to complete several surveys through the course of the semester.

Exams

There will also be three exams. Two 1.5-hour midterms will be during the normal class time, from 9-12 pm, on **Thursday, July 3rd**, and **Tuesday, July 29th**. A two-hour final exam will be given on **Friday, August 15th**, from 9-12 pm. These *may* be in the lab room, but may be in a different room.

Access to Soda Hall

Most of your work for this course will be done in class in 273 Soda. Outside of class, you may work in any EECS lab room in which a lab section is not meeting. You may also work at home, outside of lab hours. You may obtain a card key to work in Soda on weekends or late at night by going to 387 Soda Hall and filling out the relevant forms.

Working from Home

You should plan on doing most of your work in lab so that you can work with others and get help from the staff. However, after a bit of set-up on your home computer it is possible to do work from home. To do work for this course on your own computer, you will need to use a recent Firefox browser. Internet Explorer will not work (or at least work well).

This next sentence probably won't make sense and that is fine. You can run scheme by connecting to the lab machines via a secure telnet connection, or by getting a scheme environment for you home computer. Most importantly - check the course website (<http://inst.eecs.berkeley.edu/~cs3/>) for more information on getting your computer setup properly.

Grading

The various course activities will contribute points to your grade as follows.

Activity	Course points	percent of total grade
project	30	15%
all mini-projects	24 (8 each)	12% (4% each)
all other homework	Scaled to 24	12%
all on-line quizzes	Scaled to 16	8%
participation/random on-line step	6	3%
midterm exams	60 (24 and 36)	30% (12% & 18%)
final exam	40	20%

Homework

A short set of homework exercises will typically be assigned at the end of each lab. The exercises will involve writing or analyzing programs and contributing to online discussions about typical programming misconceptions. Answers to the programming exercises and contributions to the discussions will be submitted online. You should expect to put in eight to twelve hours of work per week outside of class. The summer schedule moves much more quickly than classes during the school year so it would be easy to fall behind. If you finish the online exercises early, you may leave early or work on your homework.

You are expected to keep up with the class! Homework assigned in lab is due at midnight on the day of the next lab session. For example, if it is assigned on Tuesday, it is due at 11:59 pm on Wednesday. If it is assigned on Friday, it is due at 11:59 on Monday. You will at most earn half-credit for homework turned in after the due date but within 24 hours; you will not earn any credit for any homework exercise submitted more than 24 hours late.

There will be more than 24 points worth of scaled homework points to earn; your homework score, however, will be capped at 24. As such, you can miss some homework assignments and still earn the full amount that homework can count towards your final grade.

Quizzes are online, and while they may be taken outside of the lab room, **you will receive credit for at most four quizzes taken outside of your lab section.** You will not receive any credit for quizzes taken after the lab-section in which they were assigned, whether or not you take them in the lab room or out of the lab room. As with homework assignments, there will be more than 16 points worth of scaled quiz points to earn, but at most 16 will count toward your course grade.

You will also receive a small number of points for participation and for whether you have worked on particular activities in the on-line materials. Which activities we will check on won't be known to you (or to us, beforehand), but will generally be an activity for which you are asked to answer a question or create a file. The purpose of grading this is to encourage you to keep up-to-date on the lab materials; generally, the grades for these activities will consider whether you attempted it, rather than whether you did it well. You will be docked for participation points if you miss (some) lab sessions, although we may decide to additionally consider how well you contributed to the “community” in your lab session.

Your letter grade will be determined by total course points, as shown in the table to the right. There is no curve; your grade will depend only on how well you do, not on how well everyone else does.

Incomplete grades will be granted only for dire medical or personal emergencies that cause you to miss the final exam, and only if your work up to that point is satisfactory.

What is cheating? What is not cheating?

Points	Grade
185-200	A+
165-185	A
155-165	A-
145-155	B+
135-145	B
125-135	B-
115-125	C+
105-115	C
95-105	C-
75-95	D
< 75	F

Copying and presenting another person's work as your own constitutes cheating. It will be penalized at least by a 0 on the work in question and notification of the incident to the Office of Student Conduct – but there are some gray areas in computer science because it involves so much collaboration. There are some guidelines below, but if you have any doubt ask a member of the course staff.

Exams and quizzes are open book and open note. You can not use any online resource or other human resource during exams. This one is pretty cut and dry. The more complicated ethical issue comes when working on labs, homework and projects.

Labs can be worked on in partnerships. It is acceptable to work together on one computer and retype or email the program to each other. We call this *collaboration* and not *cheating*. Labs are a time for you to learn. If you find your self copying someone else's solution without understanding it you're just hurting yourself. On lab materials if you don't understand how to solve a problem you should ask someone (student or staff) for help. You and your partner could even form a group of 4 to try to solve a tough problem. Note: On the online discussion steps you should each post your own answer.

Homework is a different story. Homework is individual and it is not okay to copy other students' code. You can talk to people about the homework problem, but you need to write every line yourself.

Some of the projects are done in partnerships. It is basically the same guidelines as homework, but now you and your partner can share code.

FYI – we have a software program that we run all of the homework and projects through that checks for cheating. We run the program with a repository of all the previous and current students' solutions to see if any of them are “too similar”. People get caught for cheating in computer science classes almost every semester. Most of the time, these students have just fallen behind. If this happens to you, instead of cheating, talk to us about your situation.

Sickness

If you get sick and think that you may fall behind on an assignment, let me know as soon as possible.

Schedule of Topics

A schedule of topics, readings, homework and lecture notes can be found online at <http://inst.eecs.berkeley.edu/~cs3/>

Alternative courses

Other courses in which you can learn to program include CS 3S, IDS 110, and CS 61A.

CS 3S is the self-paced version of CS 3. Unfortunately, CS 3S is not being offered this summer, but here is some information about CS 3S. Students may enroll for fewer than four units of CS 3S, in order to take only a portion of the course or to spread the complete course over more than one semester. (You need only three units of CS 3S to prepare for CS 61A, and two units would probably suffice.) The textbooks used in CS 3S differ from those we'll use, so if there's a chance you might want to switch, you should think about switching sooner rather than later. For further information, contact the Self-Paced Center, 642-9920, in room C10 Hearst Field Annex.

In IDS 110, students use programming tools like spreadsheets and data base managers as well as write programs. Programming in IDS 110 is done using Javascript. IDS 110 also has scheduled lab sections, held in the Tolman Microcomputer Facility in 1535 Tolman. Course staff makes an effort to get students with similar interests to work together in discussion and lab sections. IDS 110 satisfies the computing course requirement for entry into the School of Business Administration.

CS 61A is the first of a sequence of courses aimed at students with a particular interest in computer science. Its prerequisite is computing experience roughly equivalent to the first half of CS 3; thus students with no previous experience often take CS 3 or 3S to prepare for CS 61A. Scheme is also used for programming in CS 61A. If you've done more than a little programming, particularly if your experience includes exposure to recursion, you should probably take CS 61A rather than CS 3. If you are thinking of becoming a computer science major, you should think about taking CS 61A. CS 61A covers more material, and in a more rapid fashion, than CS 3.