

UNIVERSITY OF CALIFORNIA  
Department of Electrical Engineering and Computer Sciences  
Computer Science Division

CS 61A  
Summer 2011

Colleen Lewis

**CS 61A: Structure and Interpretation of Computer Programs**  
**General Course Information**

Instructor  
Colleen Lewis  
329 Soda Hall  
[ColleenL@berkeley.edu](mailto:ColleenL@berkeley.edu)

**Working from home**

If you have a home computer, you will want to get a Scheme interpreter for it. The Computer Science Division can provide you with free versions of Scheme for Linux, Windows, or MacOS. The distribution also includes the Scheme library programs that we use in this course. You can download copies of this distribution and notes about connecting from home at the following links:

<http://inst.EECS.Berkeley.EDU/~scheme>

<http://inst/connecting.html>

We will also be posting videos online and holding additional office hours to help you get your home computer set-up.

**Textbooks**

The textbook for this course is Structure and Interpretation of Computer Programs by Abelson, Sussman, and Sussman, second edition. It should be available in the textbook section of the ASUC bookstore and other local textbook sellers. You must get the 1996 second edition! Don't buy a used copy of the first edition. The book can also be read online, at

<http://www-mitpress.mit.edu/sicp/sicp.html>

In addition to the textbook, there are a lot of resources online. All labs, homework, projects, lecture notes, discussion notes, and all solutions will be posted online. In previous semesters there has been a set of unchanging reference material for the course called "Reader Volume 2". You can also access this online or get a copy from a friend that has taken the course. It has Brian Harvey's lecture notes, which is the foundation of the lecture notes this summer.

We have listed an optional text for the course, Simply Scheme, by Harvey and Wright. It really is optional! This textbook gives a slower and gentler introduction (with lots of practice problems) to the first five weeks of 61A, for people who feel swamped here.

**Laboratory and Discussion Sections**

In addition to the lectures Monday-Thursday, the course consists of two discussion sections and two lab sections each week. (You will probably also use some additional un-scheduled lab time to do homework problems.) NOTE: The first Tuesday's discussion section will meet in the lab!

Students ask whether section attendance is required or optional. Our expectation is that you will attend all class sessions. Much of the learning in this course comes from lab activities, and later assignments (including exam questions) may build on those activities. It is up to you to find out what happened at any class session that you miss.

## **Enrollment/Waitlist**

Please sign-up for a section that is not full. Telebears will be our source for what students are in each section. Please try to be in a final section on Telebears by the third day of class.

You must have a computer account on the 61A course facility. This will be distributed by your TA in the first lab. You must set up your account this week. If you are pre-enrolled but do not set up your account this week, you may be dropped from the course. The first time you log in, you will be asked to type in your name and student ID card number, if you have one. Please follow the instructions carefully. You must get your account and log into it no later than 4pm Wednesday so that we have an accurate class count. (In any case, you should be doing the first homework assignment by then.)

## **Groups**

Your first and most important resource for help in learning the material in this course is your fellow students. Your discussion section TA will assign you to a group of four students, and you will do all course activities with this group. You are responsible for helping each other learn.

# **Getting Help**

## **Tutors & Lab Assistants**

The class will have a staff of undergraduate Lab Assistants and Tutors (LAs/Tutor). Lab assistants will help during scheduled labs. Tutors will have scheduled hours to be in the lab outside of the schedule labs. Some people are both lab assistants and tutors. These hours will be posted on the class website. Whenever an LA/Tutor is in the lab you may request that s/he answers questions about the homework or programs (but not do them for you).

The instructor and the Teaching Assistants who teach the discussion sections are also available to answer questions. You may drop in during office hours, make appointments for other times, or communicate with us by email. All office hours are posted on the class webpage.

During scheduled lab sessions, only students enrolled in that particular section may use a workstation. However, if you have a question, feel free to stop in and ask a TA. At other hours, any 61A student may use the lab on a drop-in basis. (The labs on the second floor of Soda Hall are unlocked during daytime hours.)

## **Newsgroup**

We'll be using a product called "Piazza" for answering questions that are not specific to a single student. You have to register for an account with Piazza. The registration web page is

<http://www.piazza.com/class#cs61asummer2011/0>

For technical questions about the homework or about the computer facility, post your questions to Piazza, but please don't post your homework solutions before the assignment is due. You should check regularly for announcements, and this is also the best place to ask technical and administrative questions. If you have a question about code you can email all TA's and Colleen by emailing [cs61a@imail.eecs.berkeley.edu](mailto:cs61a@imail.eecs.berkeley.edu)

## **Lectures**

During lectures we'll be using iClicker technology to have students answer multiple choice questions and respond to various questions. This "lecture" won't look like other lectures; you'll be thinking, writing code, and discussing ideas with

your peers. There will be 9 points awarded for participating in lecture using the iClickers. To earn these points you must answer iClicker questions in 70% of the lectures (roughly 21 lectures). My plan is to not grade these “clicks” based upon correctness. Although if it appears that people aren’t trying to get the questions correct, I’ll enforce a threshold to later lectures. This would mean that you have to get some percent of the questions correct to earn credit for that day. You must register your iClicker using the iClicker website.

For those of you who miss lectures, or wish to see Brian Harvey explain the same topics reviewed in lecture, they are available online: <http://webcast.berkeley.edu/courses>

## Administration

### Homework Extensions

The summer goes very fast, so we have an aggressive set of homework deadlines so that you don’t fall behind. However, we know things come up and you may request 3 extensions for homework assignments during the summer. The extensions do not require a reason and provide an extension of 3 days or 72 hours. **The late homework must be turned in using 2 ways before the extension deadline. You must turn in the homework both online and in a paper copy into the homework box in 283 Soda.** You must submit the homework extension form (available from the class webpage) before the original homework deadline to be valid.

### Grade Complaints

If you have a complaint about a missing or incorrect grade on a past assignment, there is an online complaint form accessible from the course web page. All grades will be posted with 7 days of the assignment due date. This grading complaint must be made with 14 days of the assignment due date to be considered. If you do not have a grade for an assignment after 7 days, please submit a grade complaint form.

### Online Resources

There is a class web page, with online versions of some of the documents we hand out:

<http://www-inst.eecs.berkeley.edu/~cs61a>

The web page for the textbook, with additional study resources, is

<http://www-mitpress.mit.edu/sicp/sicp.html>

The optional Simply Scheme text is online at

<http://cs.berkeley.edu/bh/ss-toc2.html>

There are also web pages for the Scheme programming language:

<http://www.swiss.ai.mit.edu/projects/scheme/index.html> <http://www.schemers.org/>

Additional information to help you in studying, including hints from the course staff and copies of programs demonstrated in lectures, is available on the course web page.

### After Hours Access to Soda

271 Soda and 273 Soda are the primary lab rooms, although the CS 61A accounts can also be used from any EECS Instructional lab in Soda or Cory Hall.

The lab is normally available for use at all times, but you need a card key for evening access to the lab.

Current UCB students: If you are enrolled in the course, your Cal student ID serves as your card key and will automatically be activated for access to the Soda second floor labs (including entering the building). You do not have to do anything, unless for some reason it doesn't work, then see below.

Concurrent/other students: You can fill out an application and obtain a white card key from 387 Soda Hall (the front desk). The fee is \$20.00 cash/check, of which \$15 is refundable when you return the card.

## Homework and Programming Assignments

Each week there will be problems assigned for you to work on, most of which will involve writing and debugging computer programs. You'll work on some of these problems individually, and some in groups. These assignments come in three categories:

### Lab

Laboratory exercises are short, relatively simple exercises designed to introduce a new topic. Most weeks you'll do these during the scheduled lab meetings Monday and Wednesday, in groups of two to four students. These are ungraded and will not be turned in.

### Homework

Homework assignments consist mostly of exercises from the textbook; you'll do these whenever you can schedule time, either in the lab or at home. You may be accustomed to textbooks with huge numbers of boring, repetitive exercises. You won't find that in our text! Each assigned exercise teaches an important point. You are encouraged to discuss the homework with other students, but each of you must prepare and turn in your own solutions. (More on this later.)

The purpose of the homework is for you to learn the course, not to prove that you already know it. Therefore, the weekly homework is not graded on the correctness of your solutions, but on effort. **If you get stuck on a question**, you can still earn full credit if you indicate that you are stuck and explain (1) what you understand about the question and what it is asking, (2) what you don't understand about the question and what it is asking. You will not receive full credit unless for each assigned problem you include either your answer or these "stuck responses".

There are two ways to turn in assignments: online and on paper. The first week's homework must be turned in both online and on paper. (Part of the purpose of this assignment is to make sure you know how to print things; also, it is from the paper turn-in of this homework that the readers make lists of the students they are grading.) The remaining homework assignments must be turned in online, and may also be turned in on paper if you would like detailed comments on your work from your reader.

Most of the weekly assignments include problems labeled as "Extra for Experts." These problems are entirely optional; do them only if you have finished the regular assignment and want to do something more challenging. The assignments also contain tips for new Unix commands that we think will be helpful to you, but are not required.

Each homework is worth two points and solutions copied from someone else will receive negative four (-4) points. This includes copying from your friend who took the course last semester as well as copying from other current students. You will get negative credit for copied solutions, and repeated offenses will lead to more severe penalties. If you don't know how to do something, it's better to leave it out than to copy someone else's work.

### Projects

Projects are larger assignments intended both to teach you the skill of developing a large program and to assess your understanding of the course material. There are four projects during the semester.

The four programming projects are graded on correctness, as well as on your understanding of your solution. The first two projects will be done individually; the last two in groups of two students, but some of the work is split up so that each problem is done by one student. The groups will be chosen by the TA of your discussion section, although we'll consider requests for specific partners or unusual constraints. You must work together to ensure that both group members understand the complete results.

For the group projects, only one member of the group will submit the entire project for the group online and on paper.

A programming project that is not ready by the deadline may be turned in 24 hours late. These late projects will count for 2/3 of the earned score. There are no extensions on programming projects.

### **Solutions to class assignments**

Solutions to homework and labs are posted online each week, in the directory `~cs61a/solutions` and on the course web page, the day after each assignment is due. You should definitely read these! They discuss most problems in some depth, with alternate solutions and suggestions for thinking about similar problems. Project and exam solutions are also posted when they are available.

### **Reading**

You should try to complete the reading assignment for each week before the lecture. (Note from Colleen: More than any other course that I took as an undergrad at Cal – the readings in this class are very important!)

### **Submitting Assignments**

Everything you turn in on paper must show your name(s), your computer account(s), and your section number. Please cooperate about this; make sure they're visible on the outside of the paper you turn in, not buried in a comment in a listing. For online submissions, your name(s), computer account(s), and section number must be included in a comment at the beginning of the file.

**Paper turn-in:** (REQUIRED for HW1 and PROJECTS and EXTENSIONS) there are boxes with slots labeled by course in room 283 Soda Hall. What you turn in should include transcripts showing that you have tested your solution as appropriate. Keep your graded papers until the semester is over. You may need them in case a grade is entered incorrectly.

**Online turn-in:** (REQUIRED for all HW and PROJECTS and EXTENSIONS) You must create a directory (you'll learn how to do that in the lab) with the official assignment name, which will be something like `hw5` or `proj2`. Put in that directory all files you want to turn in. Then, still in that directory, give the shell command `submit hw5` (or whatever the assignment name is). We'll give more details in the lab.

### **Testing and Grading**

There will be two exams and one quiz. The quiz will be held during lecture on June 23<sup>rd</sup>. We'll hold the exams in the evening instead of during the lecture time. My goal will be to write one-hour tests, but you'll have two hours to work on them. The relatively large number of tests should reduce your anxiety about ruining your grade by misunderstanding any one question.

If it were up to me, we wouldn't give grades at all. Since I can't do that, the grading policy of the course has these goals: It should encourage you to do the course work and reward reasonable effort with reasonable grades; it should minimize competitiveness and grade pressure, so that you can focus instead on the intellectual content of the course; and it

should minimize the time I spend arguing with students about their grades. To meet these goals, your course grade is computed using a point system with a total of 300 points:

Quiz	5
Exam 1	40
Exam 2	40
Final	100
Homework	26 (2 pts/hw)
Project 1	15
Project 2	15
Project 3	25
Project 4	25
Participation (iClickers)	9

Each letter grade corresponds to a range of point scores:

A+	288-300	A	281-287	A-	274-280
B+	258-273	B	242-257	B-	226-241
C+	219-225	C	212-218	C-	205-211
D+	195-204	D	185-194	D-	170-184

This grading formula implies that there is no curve; your grade will depend only on how well you (and, to a small extent, your group partners) do, and not on how well everyone else does. (If everyone does exceptionally badly on some exam, I may decide the exam was at fault rather than the students, in which case I'll adjust the grade cutoffs as I deem appropriate. But I won't adjust in the other direction; if everyone gets an A, that's great. Any such adjustment will happen at the end of the semester, after all the grading is done, and will be based on the total grade distribution. Often a hard question on one exam is balanced by an easy question on another exam, even though we don't try to write either too-easy or too-hard questions.)

If you believe we have incorrectly graded an exam, first be sure that you understand the solutions and grading standards that we'll post online soon after the exam. If your paper was incorrectly graded according to those standards, return it to your TA and staple an explanation of your complaint to the front of the exam. By University policy, final exams may not be re-graded. They may be viewed at times and places to be announced.

If you believe any assignment has been incorrectly graded, or your grade isn't posted within one week of turning it in, you should use the online grading complaint form reachable from the class web page. You must do this within 2 weeks of the original assignment deadline.

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

### **Cooperative Learning**

With the obvious exception of exams, we encourage you to discuss all of the course activities with your friends as you are working on them. Parts of some midterm exams will be done in groups of four; this will be explained further when the time comes.

Each student must solve the weekly homework assignments individually; it is by struggling with the homework problems that you learn the course material! However, before you develop your own solution to each problem you are encouraged to discuss it with other students, in groups as large or small as you like. When you turn in your solution, you must give credit to any other student(s) who contributed to your work. Working on the homework in groups is both a good way to learn and a lot more fun! Although the homework is graded on effort rather than on correctness, if you take the opportunity to discuss the homework with other students then you'll probably solve every problem correctly. Similarly, each student should solve each problem in the scheduled lab activities, but you are welcome to discuss your efforts with your neighbors in the lab.

In addition to the 300 points listed above, your working group can earn up to six bonus points if every group member works to ensure the success of every other member, as follows. For the second exam and the final, every member of your group will get two bonus points if your original assigned group is still intact and the person in your group with the lowest score on the previous midterm improves by at least five points (for the final, appropriate scaling is done). But you don't get the bonus points if the lowest-scoring student on the previous exam has dropped the course or changed groups. The intent of this rule is to encourage the group to study together and to take active responsibility for everyone's learning.

Working cooperatively in groups is a change from the traditional approach in schools, in which students work either in isolation or in competition. But cooperative learning has become increasingly popular as educational research has demonstrated its effectiveness.

### **Cheating**

The EECS Department Policy on Academic Dishonesty says, "Copying all or part of another person's work, or using reference materials not specifically allowed, are forms of cheating and will not be tolerated." The policy statement goes on to explain the penalties for cheating, which range from a zero grade for the test up to dismissal from the University, for a second offense.

In my experience, nobody begins the semester with the intention of cheating. Students who cheat do so because they fall behind gradually, and then panic at the last minute. Some students get into this situation because they are afraid of an unpleasant conversation with an instructor if they admit to not understanding something. Please ask for help early.

## **Questions and Answers**

### **Q: Is it true that 61A is the weed-out course for wannabe CS majors?**

A: No. The CS major is not overcrowded these days, and we have no interest in weeding anyone out. This class is hard, but we want everyone to be very successful!

### **Q: I heard that CS61A will be taught with Python in the future, is this true?**

A: Yes – this may be the last semester that the course is offered in Scheme. It is a beautiful language using an exceptional textbook and we are certain you'll get a lot of out of the course using Scheme.

### **Q: I need special facilities or arrangements to do the course work. What should I do about it?**

A: The Disabled Students Program (DSP, ext. 2-0518) provides documentation of required accommodations to instructors. Please have the Disabled Students Program send us documentation ASAP so that we can make appropriate arrangements.

### **Q: Can I form a group with a student in another section?**

A: Nope. One purpose of the scheduled lab meetings is to ensure that your group can spend some time working together with your TA available to help. If you want to be in the same group with a friend, arrange your schedules so that you can be in the same section.

**Q: My project partner never does any work/I've had trouble contacting them. What should I do about it?**

A: Talk to your TA as soon as possible!!!!

**Q: I forgot my password. (Or: It won't let me log in.)**

A: Go to the Instructional sys admin staff in 333 Soda, 378 Cory or 386 Cory. Bring your initial class account form or student ID card. Don't ask for another account, especially if you already have some work graded under the old account!

**Q: What is this course about?**

A: The CS 61 series is an introduction to computer science, with particular emphasis on software and on machines from a programmer's point of view. This first course concentrates mostly on the idea of abstraction, allowing the programmer to think in terms appropriate to the problem rather than in low-level operations dictated by the computer hardware. The next course, CS 61B, will deal with the more advanced engineering aspects of software—on constructing and analyzing large programs and on techniques for handling computationally expensive programs. Finally, CS 61C concentrates on machines and how they carry out the programs you write.

In CS 61A, we are interested in teaching you about programming, not about any particular programming language. We consider a series of techniques for controlling program complexity, such as functional programming, data abstraction, object-oriented programming, and query systems. To get past generalities you must have programming practice in some particular language, and in this course we use Scheme, a dialect of Lisp. This language is particularly well-suited to the organizing ideas we want to teach. Our hope, however, is that once you have learned the essence of programming, you will find that picking up a new programming language is but a few days' work.

**Q: Do You Belong Here?**

A: In the past we did not impose any programming-related prerequisites for admission to 61A. However, in recent years we have found that 80% to 90% of 61A students have had significant prior programming experience, and that students without such experience are at a disadvantage. You certainly have adequate background for this course if you are familiar with the idea of recursion: a procedure invoking itself as a sub procedure. There is no need for you to be familiar with any particular programming language, although if all of your experience has been in BASIC then you probably haven't used recursion. If you've taken the CS Advanced Placement AB course in C++ or Java, you are certainly ready for 61A.

If you don't feel ready for 61A, we recommend that you take CS3L which will introduce programming and recursion using scheme. If you wait until the fall or spring you could take CS 10, which is an introduction to computer science for non-majors including Scratch programming, or CS 3S, a self-paced programming class for people who are pretty sure they will be CS or EECS majors. Both are designed to serve as preparation for 61A. You could then take 61A later. Note: in the fall of 2011 and spring of 2012, CS61A will be taught using the programming language Python.

If you have substantial prior programming background, you may feel that you can skip 61A. In most cases we don't recommend that. Although 61A is the first course in the CS sequence, it's quite different from most introductory courses. Unless you have used this same textbook elsewhere, I think I can promise that you won't be bored. Perhaps your prior experience will allow you to skip 61B or 61C, which are more comparable to courses taught elsewhere.