

Welcome to CS 61BL, “Data Structures and Programming Methodology”

Instructor: Colleen Lewis

This overview contains information about textbooks, labs and discussions, grading, and various policies that will be administered in this course. Please read it carefully.

What’s this course all about?

CS 61A covered high-level approaches to problem solving, providing you with a variety of ways to view programming problems (in terms of functions, objects, or rules). In CS 61BL, we move to a somewhat more detailed level of programming. As in 61A, the correctness of a program is important; in addition, however, we’re concerned with a program’s efficient use of time and memory resources. Much of 61BL will be devoted to the tradeoffs in time and memory that arise from a variety of methods for structuring data.

How is summer session different?

The summer version of the course covers the same material that is covered during the regular 14 week semester. The summer session runs for a total of 8 weeks, which is much shorter! CS61B is a very demanding course even for a full length semester so expect there to be a lot of work.

During the semester long version of CS61BL you would have had two – 3 hour labs a week plus 1 hour of lecture. This summer you will have four - 3 hour labs a week and two – 1 hour lectures a week.

What’s the difference between CS 61BL and CS 61B?

CS61B is the “normal” course with a sprinkling of lecture, discussion and lab. CS61BL shifts the focus from lectures to lab. This summer we’re doing the lab-based version of CS61B, CS61BL, so each week there will be four – 3 hour labs and two - 1 hour lectures.

Why are we doing the lab-based version?

- A common theme in computer science education research is that people learn to program by actually programming, not by listening to lectures about programming.
- Software engineering job interviews ask questions that almost exclusively come from the course content of CS61B. In this course we don’t only want you to master the material; we want you to master being able to *explain* the material and the lab environment can provide what we believe to be crucial practice.
- Sometimes you just need an extra pair of eyes (or ears) to help you get past a bug that might have wasted hours of your time if you were alone. In the lab based course you do more of your work with the support of your peers and TA.
- The lab time provides the time for TAs and lab assistance to provide feedback on daily quizzes. So the moment you start to feel that you’re getting behind or the moment your quiz answer suggests that you’re behind – we’ll be there for you.

If you think you’re going to really miss the lectures, Professor Jonathan Shewchuk has wonderful lectures available online for CS61B.

Lecture Notes: <http://www.cs.berkeley.edu/~jrs/61b/>

Lecture Videos: http://webcast.berkeley.edu/course_details.php?seriesid=1906978343

Staff

Instructor	Colleen Lewis		colleenl@berkeley.edu
TAs	Jonathan Kotker	8 - 11	cs61b-ta@imail.eecs.berkeley.edu
	Kaushik Iyer	11 - 2	cs61b-tb@imail.eecs.berkeley.edu
	David Zeng	2 - 5	cs61b-tc@imail.eecs.berkeley.edu
	George Wang	6 - 9	cs61b-td@imail.eecs.berkeley.edu
Reader	Angela Juang	All sections	cs61b-ra@imail.eecs.berkeley.edu

Books

There are three required texts for the course. They are available at the campus bookstores:

- **Head First Java**, second edition, by Kathy Sierra and Bert Bates (O'Reilly, 2005);
- **Pragmatic Unit Testing in Java with JUnit**, by Andrew Hunt and David Thomas (The Pragmatic Bookshelf, 2004); and
- **Data Structures and Abstractions with Java**, second edition, by Frank M. Carrano (Prentice Hall, 2007).

Data Structures and Abstractions with Java also comes in an online version. For roughly half the price you can get a five-month subscription to a "WebBook" that can be searched, annotated with notes, printed, and bookmarked; see <http://www.coursesmart.com/search> and search for the title of the book or search for the ISBN: 0-13-237047-6

Reading Assignments

All reading assignments will be posted online. Additional online materials will be assigned as reading.

Newsgroup

The newsgroup is required reading. Please post all questions to the newsgroup instead of emailing. That way the instructor, TAs, lab assistants and other students might be able to answer your question as quickly as possible. Note: You are never allowed to post code that is part of a homework or project on the newsgroup. Please submit a request online to be able to view the newsgroup as soon as possible: http://groups.google.com/group/cs61b_su09

Lectures

There will be 2 lectures per week. These lectures will take place on Tuesdays and Thursdays from 5:10-6:00pm, in 10 Evans. Lecture notes will be posted online after the lecture

Labs

There will be 32 labs, held in 275 Soda. During each lab there will be a 30 minute discussion section led by your TA. During the discussion section you are expected to stop working on your lab, homework, and projects. **During these discussion sections there will be a lot of talking and collaboration, however for any non-discussion section related conversation you must leave the room.**

Lab materials are all available online from ucwise.org

Quizzes

There will be 30 quizzes, one quiz during every lab day except the first and last. Each day will begin with the quiz. You should spend no more than 15 minutes on the quiz. You must be present in lab to take the quiz and earn points. Quiz scores are capped at 70%. This allows you the freedom to miss a few quizzes if needed and allows us the freedom to make the quizzes a bit on the hard side so as to help you and your TA figure out where you need help.

Exams

There will be two midterms and one final

- **Midterm 1:** Tuesday July 7th, 5-6pm in 10 Evans
- **Midterm 2:** Tuesday July 28th, 5-6pm in 10 Evans
- **Final:** Thursday August 13th, 5-8pm in 10 Evans

Review Sessions

There will be 3 review sessions

- **Review 1:** Sunday July 5th, 1-4pm in 306 Soda
- **Review 2:** Saturday July 25th, 1-4pm in 306 Soda
- **Review 3:** Sunday August 9th, 1-4pm in 306 Soda

Projects

There will be three projects

- **Project 1** (individual): due Monday July 13th, 10pm
- **Project 2** (with 0 or 1 partner): due Wednesday July 22nd, 10pm
- **Project 3** (with 1 or 2 partners): due Tuesday August 11th, 10pm

Note – No individual submissions will be accepted for Project 3.

Homework

There will be homework assigned during labs and will be submitted online. Homework when assigned in lab will be due before - not your next lab – but the following lab (see below).

Assigned:	Due:
Monday during lab	Wednesday before your lab
Tuesday during lab	Thursday before your lab
Wednesday during lab	Monday before your lab
Thursday during lab	Tuesday before your lab

All homework will be individual. There will be no late homework accepted.

Office Hours

TAs will not hold office hours, however are available during lab times. If you need help, TAs, lab assistants and other students are available in 275 Soda from 8-5pm and 6-9pm Monday through

Thursday. However, during the lab's discussion section the TA and other students will not be available to help you. If there are any lab assistants, they can help in a neighboring classroom.

Colleen will be holding her office hour on Wednesday from 5-6pm in 275 Soda. Additional office hours will be announced via the newsgroup. Please email Colleen if you would like to schedule additional office hours (ColleenL@berkeley.edu).

Grading

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

Assignment category	# points	Percent of total grade
All projects	36	18%
All other homework scaled to:	24	12%
All quizzes scaled to:	20	10%
Midterm 1	24	12%
Midterm 2	36	18%
Final	60	30%

Your letter grade is determined by total course points, as shown in the table below: In other words, there is no curve; your grade will depend only on how well you do, not on how well everyone else does.

Points	Grade
190-200	A+
180-190	A
170-180	A-
160-170	B+
150-160	B
140-150	B-
130-140	C+
120-130	C
110-120	C-
100-110	D+
90-100	D
80-90	D-
<80	F

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.

You can view the scores you have received by typing **glookup** in an `xterm` window.

Cheating

Policy on collaboration and cheating: don't cheat.

Copying and presenting another person's work as your own constitutes cheating. Consistent with departmental policy, incidents of cheating on homework or labs will result in a negative grade on that assignment, while cheating on projects or exams will result in a failing grade in the course. All incidents of cheating will be reported to the Office of Student Conduct where records of academic misconduct are kept throughout your undergraduate career.

On every homework assignment and project we will be running cheating detection software to compare your solution with the solution of your classmates and with all previous submissions. If you're falling behind at all – please talk to Colleen or your TA!

Sometimes it's hard to draw the line between cheating and appropriate collaboration. Here are some rules of thumb:

- **Obvious rules:** You can't get a solution from the internet by any method. Don't copy a current or previous student's solution.
- **No code rule:** You should never have any part of a current or previous student's homework or project solution in your possession, either electronically or in hardcopy form.
 - **Lab collaboration clarification:** In lab you will write a lot of programs that you will never turn in for homework or a project. For a program that is not turned in, we encourage you to write code with a partner on one computer and then share that collaboratively written code. Always check the day's homework assignment to make sure you don't need to turn that code in. If you do need to turn it in, you can talk, but you can not share code.
- **Reusing code:** You may reuse code that you have written. The first time you are writing the code if it is for a homework or project you **MUST** write it yourself. If not, you may work together with a partner. If you would like to reuse some code that you and your partner wrote in a regular lab that was not part of a homework or project, you may reuse that code. Please make a note whenever this is the case.
- **Looking at code rule:** You can help a classmate with bugs in their homework or projects and in doing so you can look at their code. Don't copy their code that you see – all work must be your own and all work must be their own.
 - **Don't show your own:** If you're helping a classmate find a bug, it is **NOT** okay to show them your code – whether it is working or not. You should focus only the buggy code in question.
 - **Debugging clarification:** You can help someone if they have a logical error in their code. You should:
 - Never touch their keyboard or mouse.
 - Ask them lots of questions! Examples:
 - If they have a run time error:
 - What is happening on the line indicated by the run-time error?
What are you trying to do?

- How could that error happen on that line? What would need to be true? How could those conditions be met?
- If the program doesn't do the right thing:
 - What test case(s) demonstrates the error?
 - Can you come up with a simpler test that demonstrates the error?
 - Can you trace through by-hand a simple example that demonstrates the error?
 - Have you run the debugger on the simple test to see when the error is introduced?
 - Have you tested the sub-procedures to make sure that they work?
- **Compiling clarification:** If their code doesn't compile you can help! You should:
 - Never touch their keyboard or mouse.
 - Try to help them figure it out for themselves. You should just be *helping*.
 - Talk about each syntax error to make sure that it makes sense why the compiler was complaining.
- **When in doubt – ask rule:** If you are not sure whether a particular interaction is appropriate, talk to Colleen Lewis or your lab TA before you submit the solution.
- **Give credit rule:** If you receive a significant idea from someone else, clearly acknowledge that student in your solution. Not only is this a good scholarly conduct, it also protects you from accusation of theft of your colleagues' ideas.