Welcome to CS 61BL, "Data Structures and Programming Methodology".

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. (The third course in the sequence, CS 61C, continues the journey from high-level programming to the details of the machine level, covering the basics of computer architecture and revealing aspects of the computer that programming environments hide from you.)

The 61ABC sequence arose from a desire to enhance the connection between CS 61A and the data structures material. In this course we will make frequent reference to material covered in CS 61A: recursion, linked data structures, and object-oriented programming.

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

The traditional version of CS 61B has loosely coordinated lecture, discussion, and lab sections. More recently, we have introduced a *lab-based* version of the course (the "L" in 61BL stands for "lab") that trades lecture and discussion time for hands-on lab time. The two versions cover equivalent content; students do about the same amount of total work in each version. In the lab-based version, we integrate the various parts of the course better and supervise your work more closely; as a result, we are better able to provide help when you're confused.

Staff

The instructor is Mike Clancy, clancy@cs, 779 Soda, 642-7017 (his administrative assistant is Audrey Raya in 385 Soda). The teaching assistants are Andrew Dai (adai@berkeley), Mike Demmer (demmer@cs), Jiahan Jiang (jiahan_jiang@berkeley), Leonard Wei (lenwei@berkeley), and Wei Zheng (zhengwei@eecs). There will also be lab assistants working with us during the lab sections, and readers to grade your homework. Detailed information on contacting staff will be available shortly.

Books and other course material

There are four required texts for the course. Three 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).

Head First Java has been used in CS 61B for several semesters, so used copies should be readily available. *Data Structures and Abstractions with Java* also comes in an online version. For roughly half the price of the hard-cover version, you can get a five-month subscription

to a "WebBook" that can be searched, annotated with notes, printed, and bookmarked; see http://vig.prenhall.com/catalog/academic/product/0,1144,013237045X-AE,00.html for more details.

The fourth text is named "CS 61B Readings", available at Copy Central, 2483 Hearst. We will be using the Readings, *Head First Java*, and *Pragmatic Unit Testing* the first month or so of the semester, and *Data Structures and Abstractions with Java* later on.

Should you need a guide to the UNIX operating system, we suggest the book *Your UNIX: The Ultimate Guide*, by Sumitabha Das, available at a bookstore. The Computer Science Undergraduate Association has several UNIX help sessions scheduled:

Tues, Jan 16	7–8pm	Introduction to the instructional computing operation
Wed, Jan 17	6–7pm	Introduction to UNIX
Thurs, Jan 18	7–8pm	Introduction to emacs
Mon, Jan 22	6–7pm	Intermediate UNIX
Tues, Jan 23	7–8pm	Introduction to vi
Wed, Jan 24	6–7pm	Introduction to shell scripting
Thurs, Jan 25	7–8pm	Introduction to Python

All are in 306 Soda. Check the URL http://www.csua.berkeley.edu/help-sessions for further details.

Class activities and scheduling

There will be a one-hour lecture each week on Wednesdays from 5 to 6pm. (The lectures from CS 61B were webcast last semester and will be again this semester. The webcasts, available at http://webcast.berkeley.edu, may be a useful resource.) In-class exams on Wednesday, February 14 and Wednesday, April 4 will extend into the 6-7pm hour.

You are each assigned a lab section that you are to attend each week. Available sections are TuTh 8-11am (run by Andrew Dai), TuTh 11-2pm (Jiahan Jiang), TuTh 2-5pm (Mike Demmer), TuTh 5-8pm (Wei Zheng), WF 10-1pm (Mike Demmer), and WF 1-4pm (Leonard Wei). Some of the sections are fully enrolled; open slots in these sections will be filled according to waiting list position. You may attend a section to which you're not assigned only with the permission of both the relevant lab t.a.s.

For most of the semester, the typical lab period will involve a variety of activities, all provided online. It will start with a short quiz based on topics covered on the homework or in the preceding lab; each start-of-period quiz will count toward your course grade. It may conclude with another quiz covering the material you just worked with; each end-of-period quiz will function as a diagnostic to give you feedback on what you don't fully understand. In between, you'll be reading, experimenting, brainstorming, evaluating each other's ideas, and sometimes working with partners.

A short set of homework exercises will be often 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. Solutions to the programming exercises and contributions to the discussions will be submitted online. There will be three larger

project assignments during the semester, to which some of the lab meetings will be devoted. (The first two projects will be individual efforts; for the third, you'll be allowed to work in partnership.) You should expect to put in at least four or five hours of work per week outside of class. If you finish the online exercises early, you may leave early or work on your homework.

There will also be three exams. Two will be in class, on Wednesday, February 14 and Wednesday, April 4, from 5 to 7pm. (Note the extra hour.) The other will be the final exam, Thursday, May 17 from 5 to 8pm (exam group 18).

The programming tools and course material used in CS 61BL were devised by a research group of computer science and education researchers. To determine the effectiveness of these tools and material, we will gather data on your background and performance, via questionnaires, interviews, and analysis of your work.

Computing

Most of your work for this course will be done in class in 275 Soda. Outside of class, you may work in any EECS lab room in which a lab section is not meeting.

To access the CS 61BL Course Portal, `http://spring07.ucwise.org`, you should use a recent version of the Firefox browser, with a plugin that supports Java 1.5. Neither earlier versions of Mozilla nor any version of Internet Explorer will work well. There are a variety of free Java programming environments available, e.g. JBuilder, DrJava, NetBeans, and Eclipse. We will use Eclipse in lab. You may download it for your own computer from `www.eclipse.org/downloads`.

You will each be given an account on the EECS instructional computers. Account forms will be handed out in your first lab section. You should change your password after receiving your account. Initial access to the CS 61BL Portal also requires an enrollment code, which t.a.s will provide in your lab section.

The Soda labs are open from 7:30am to 6pm Monday through Friday. Outside of these hours the doors to the building and lab are locked, and you will need to obtain card key access to use the labs. Current students with CAL 1 identification cards can enable them for lab access by visiting 387 Soda. Students registered through U.C. Extension should apply at 387 Soda for a white card key; the fee is $20 (with $15 refunded when you return the card.

Grading

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

activity	*course points*	*% of total grade*
all projects	36	18%
all other homework	scaled to 24	12%
all quizzes	scaled to 20	10%
first midterm exam	24	12%
second midterm exam	36	18%
final exam	60	30%

Your quiz score will be capped at 70% of the maximum total points, then scaled to 20. Similarly, your homework score will be capped at 90% of the maximum total points, then scaled to 24.

You are expected to keep up with the classwork. There will occasionally be time devoted in lab to helping you catch up or solidify your understanding of the material. Homeworks assigned in one lab section are due at the start of the next section unless otherwise specified. You will be expected to submit homework on time, and to take quizzes on the day they are assigned. Quizzes are online, and thus may be taken outside of the lab room. You will receive credit for at most four quizzes taken outside of your lab section. Any quiz assigned in one lab section must be completed by the start of the next lab section to earn any credit.

Your letter grade is determined by total course points, as shown in the table below:

points	180–200	160–180	150–160	140–150	130–140	120–130
grade	A+	A	A–	B+	B	B–
points	110–120	100–110	90–100	70–90	<70	
grade	C+	C	C–	D	F	

In other words, 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.

Approximate topic/activity schedule

Here is a tentative schedule of topics to be covered, project due dates, and exams. Titles of the four course texts are abbreviated as follows:

- *Head First Java* = HFJ
- *Pragmatic User Testing* = PUT
- *Data Structures and Abstractions with Java* = DSAJ
- *CS 61BL Readings* = 61BLR

week of …	*events*	*topics*	*reading*
January 15	Martin Luther King holiday (Monday)	introduction to Java; flow-of-control; testing	HFJ chapters 1-2; 61BLR, “CS 61A Scheme to CS 61B Java”; DSAJ sections 10.1-10.14 (recursion review)
January 22		variables; objects; references; arrays; test-driven development; JUnit	HFJ chapters 3, 4, and 9 except material on inheritance in pages 250-255; PUT chapters 1 and 2; DSAJ sections 10.15-10.19; 61BLR, “Boxes and arrows”
January 29		for loops; more on arrays and testing; java.util	HFJ chapters 5, 6, and 10, and Appendix B #7 and #2; PUT sections 3.1-3.3 and chapters 4 and 5; DSAJ chapters 2 and 3

week of …	*events*	*topics*	*reading*
February 5		sorting; inheritance, polymorphism, and interfaces; exceptions	HFJ chapters 7 and 8, pages 250-255 in chapter 9, and Appendix B #4; DSAJ chapter 11
February 12	exam 1 (Wednesday)	big-Oh notation; linked lists; comparative implementations	61BLR, "Notes on linked data structures"; DSAJ chapters 3-8, 13, 14, and sections 10.20-10.21
February 19	Presidents' Day holiday (Monday)	stacks and queues; trees and tree iterators	HFJ chapter 11; DSAJ chapters 9, 16, and 25, and sections 10.22 to the end of chapter 10
February 26	project 1 due (Monday)	more on trees	DSAJ chapters 21, 22, and 26
March 5		hashing	HFJ chapter 16; DSAJ chapters 17, 18, and 27
March 12		more on hashing; search trees; project work	DSAJ chapters 19 and 20
March 19	project 2 due (Monday)	priority queues	
March 26	spring break		
April 2	exam 2 (Wednesday)	graphs; more on sorting	DSAJ chapters 23, 24, and 28
April 9		project work	DSAJ chapters 30 and 31
April 16		balanced trees, tries; project work	DSAJ chapter 12
April 23		project work; review	DSAJ chapter 29
April 30	project 3 due (Friday)		

Policy on collaboration and cheating

Copying and presenting another person's work as your own constitutes cheating. Cheating directly affects the reputation of the Department and University and lowers the morale of other students. Consistent with departmental policy, incidents of cheating on homeworks 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.

Sometimes it's hard to draw the line between cheating and appropriate collaboration. Obviously wrong is getting a "homework service" to write your program for you, or finding a solution on the Internet and submitting it as your own. On the other hand, providing suggestions to a classmate about the meaning of a question and offering advice about the likely meaning of a compiler error message are examples of interaction that we encourage. The various collaboration activities in CS 61B may complicate matters further. Some useful rules of thumb, however, are the following:

- There is no reason that you should ever have another partnership's solution in your possession, either electronically or in hardcopy form. (We will call this the "no code rule".)
- If you are not sure whether a particular interaction is appropriate, talk to Mike Clancy or your lab t.a. before you submit the solution.
- 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.

Fortunately, we have found in the past that the close interaction between lab instructor and students in a lab-based class serves to support student learning (reducing the need to cheat) and to convince students both that the instructor cares what they learn and that solutions that differ significantly from students' earlier work will be detected.

Other courses

If you're interested only in learning Java, the self-paced course CS 9G ("Java for Programmers") may be more appropriate for you. CS 9G is not a prerequisite for CS 61BL, nor does it satisfy any requirements that CS 61BL does.

CS 9E ("Personal UNIX") is another self-paced course of interest. It covers UNIX use in more depth than does any other course currently offered in the Computer Science Division.

Both courses (and the other CS 9 courses) are 1 unit, and must be taken P/NP. There are no lectures other than the identical orientation meetings Tuesday, January 16 and Wednesday, January 17, both 4 to 5pm in 430 Soda.

Those of you who have had a data structures course that covered most but not all of CS 61B are eligible to take CS 47B, a one-unit course that completes the material of CS 61B and satisfies all requirements for CS 61B. CS 47B is also self-paced, graded. Programming is done in Java.