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

CS61B **M. Brudno**
Summer 2000

CS 61B: Data Structures and Advanced Programming
General Course Information*

Introduction

Welcome to CS 61B. 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. CS 61A covered high-level approaches to problem-solving, providing you with a variety of ways to organize solutions to programming problems: as compositions of functions, collections of objects, or sets of rules. In CS 61B, we move to a somewhat more detailed (and to some extent, more basic) level of programming. As in 61A, the *correctness* of a program is important. In CS 61B, we're concerned also with a program's *efficient use of time and memory resources.* Much of 61B will be concerned with the tradeoffs in time and memory for a variety of methods for structuring data. We'll also be concerned with the engineering knowledge and skills needed to build and maintain moderately large programs.

Instructor: Michael Brudno, 784 Soda Hall, 642-1898, brudno@cs.berkeley.edu
My office hours after the first week will be Monday and Thursday 1-3 in my office or a nearby alcove, and also by appointment.

Teaching Assistants and other staff: The teaching assistants this semester are Billy Chen (cs61b-tf@cory.eecs), Alice Gutman (cs61b-ta@cory.eecs), Geoff MacDonald (cs61b-tc@cory.eecs), John (Jack) Sampson (cs61b-tb@cory.eecs), Jeff Schoner (cs61b-td@cory.eecs) and Christopher Shaw (cs61b-te@cory.eecs). Their office hours will be posted later on the course website. There will also be readers grading the homeworks and staffing the labs sections. We'll post their names, e-mail addresses, and hours as soon as they're finalized.

Your "official" TeleBEARS section doesn't really mean anything in terms of your enrollment in this class. The most important thing is that you find a section that has space and go it for the rest of the semester. Once you've chosen a section, you need to stay in that section unless you get specific approval from the TAs involved. There is a limit of 35 people

*With contributions by Brian Harvey, Mike Clancy, Katherine Yelick, John Canny, and Jeffrey Forbes.

in any section, despite what Telebears says. No one will be allowed to enroll in the sections which have more then 35 students. Please don't even think about informing TeleBEARS if you change sections; it will only cause trouble, and it makes no difference for my records. Do inform the TAs involved.

On-line Resources

The course home page provides one-stop shopping for course information. All the handouts, homeworks, labs, staff contact information, etc., will be posted there. The home page is

```
http://www-inst.eecs.berkeley.edu/~cs61b
```

If you're logging in over a terminal that can't run Netscape or Explorer, you can still access the page text-only using lynx. Type `lynx` followed by the URL above.

The course newsgroup is `ucb.class.cs61b`. For most questions about the course, the newsgroup is the right place to ask them. All the course staff read it regularly, so you will get a quick answer. That way, other students benefit by seeing the question and the answer. Don't forget to check the newsgroup before asking your question, just in case someone else has posted it.

The e-mail address `cs61b@cs` will send a message to the entire staff. You can use it for correspondences that you don't want to send to the newsgroup. Again, all the staff read it, so you will usually get a quick reply. If you send a question that is of general interest, we may post the response on the newsgroup (we will keep personal information out of it, of course). To talk with us, the best way is to come during regular office hours (posted on our doors as well as in the home page). Many of us are available at other times by appointment. Please don't be shy; web pages, e-mail, and news are useful, but its still much easier to understand something when you can talk about it face-to-face.

When logged into our instructional systems for CS61B work, please make sure that you are using the standard configuration for the class—that is, the files `.cshrc`, `.login`, `.emacs`, etc., that should have been in your accounts initially. If you must modify these, we suggest that you continue to have them read our scripts from the `~cs61b/adm` directory, so that we can easily propagate corrections to you. In any case, if you modify these files, you are on your own.

One of the advantages of using Java in the course is that Java programs are highly portable (at least the kind we'll be writing). If you have an installation of Symantek's Cafe, Microsoft's J++, or Sun's JDK (Java Development Kit) at home on your PC or Mac, feel free to use it to develop solutions to programming problems. We will be using version 1.2 (aka 2) of the language; if you have a 1.0 or a 1.1 installation, some things will not work. The modified version of Sun's debugger that we use—called `gjdb`—is only likely to work on JDK 1.2; in particular, don't expect it to work at all on a non-Sun Java implementation. You will need to download some `.class` files from us. Details will appear on the class home page. The JDK (which we use in class) is free, and may be downloaded from Sun.

Background Knowledge

Some of you may have thought that the stuff you learned in CS 61A was mere esoteric fluff inexplicably thrown at you to weed out the faint of heart. Not true. In fact, although the syntaxes of Java and Scheme are enormously different, the underlying computational models are surprisingly similar. You will find that almost all the "big ideas" you see in Java had their analogues in what you learned in CS 61A (indeed, one self-test of your understanding of the course material in CS 61B is to check that you see all the similarities). This course will assume you are familiar with CS 61A, and there will be some references to the 61A textbook (Abelson, Sussman, and Sussman). If you haven't taken 61A, you may be confused sometimes, and you should make sure you review a copy of Abelson, Sussman, and Sussman early on in the semester.

All the instructional machines for this course will be running various flavors of the Unix operating systems, and it's essential that you become familiar with it. A written introduction "EECS Instructional Users Guide to Unix" by Sharon Taylor is available from Copy Central, either on Euclid or on Bancroft. Another good introduction is "A Practical Guide to The Unix System" by Mark Sobell (Benjamin Cummings pub. 1995) available at bookstores. There's also various online help available which we'll try to put links to on our web page.

Is this the right course?

This is a course about data structures and programming methods. It happens to also teach Java and a bit of C++, since it is hard to teach programming without a language. However, it is not intended as an exhaustive course on Java, the World-Wide Web, creating applets, user interfaces, graphics, or any of that fun stuff. Some of you may have already had a data structures course, and simply want to learn Java or C++. For you, a much better choice would be self-study, or (for C++), CS 9F "C++ for programmers". This is a one-unit self-paced course. It will teach you more of what you want to know in less time. There is no enrollment limit for that course, and you work through it at your own pace after the first and only lecture.

Getting into the course

I expect to enroll all students currently enrolled or wait-listed. We are putting a maximum of 245 students for the class. If you are wait-listed but meet the prerequisites (CS61A), then go to lab sections, pick up account forms, do homework, and attend lectures as if you were in the course.

Course Materials

There are two textbooks for this course: *Data Structures & Algorithms in Java* by Goodrich & Tamassia (John Wiley & Sons, 1998); and *Java 2 by Example* by Geoff Friesen (Macmillan, 2000). Both books are available in the ASUC bookstore. A reader containing various useful

documentation and supplementary course material is available from north-side Copy Central at 2483 Hearst. The same information as in the reader is also be available on-line.

There will be other handouts during the semester, and it may be necessary to collect a small fee later in the semester for copying charges. We will keep electronic copies of all handouts, including tests and homework assignments, in links from the course home page. You can also get to them directly from an instructional Unix machine in the directory `$master/handouts`.

Enrollment: Discussion Sections, Tutorials, and Labs

There will be two discussion sections per week and a tutorial. Discussion sections are groups of approximately 30 students where the TA will highlight important ideas from the current material. Tutorials are groups of no more than 10 students where a TA will lead you interactively through examples, give hints on parts of the homeworks and projects, and ask you questions about the material. Tutorials are a chance for you to get individual attention and have your questions answered. Tutorials are also a chance for us to see how you're keeping up in the class and how the work is being distributed between project partners. Attendance and participation at tutorials is *mandatory* and graded. You should also plan on attending the discussion sections. Because of the rapid nature of the course sections will often *supplement* the lectures rather than repeat them. Tests will be returned in section. There will also be lab time a s according to the schedule of classes. In that time, the lab will be staffed by readers/lab assistants. Attendance at labs is purely optional except for the first session. We may staff the lab at other times other than scheduled lab sections. Check the home page for updates on this.

Computer Accounts

The CS 61B scheduled labs will all be held in 275 Soda, which contains Intel workstations running Solaris. At other times, you can use any computer that is not being used for another scheduled lab. CS 61B has priority for 275 Soda 24 hours a day. Soda Lab times are normally posted outside each computer room. You will receive a computer account form during your first lab (this week). If you can't make that lab, a few extra forms will be available from me in my office. Information on computer facilities, dialup numbers, and so forth is available from the EECS instructional home page `http://www-inst.eecs.berkeley.edu`.

Schedule

The syllabus will be kept online at

`http://www-inst.eecs.berkeley.edu/~cs61b/syllabus.html`

It gives the due dates for homeworks and projects along with the dates for the midterms and final. Also, it has a very tentative schedule of material to be covered. CS 61B moves rather

quickly normally. In the summer session, the course can move at breakneck speed It is very important that you keep up to date.

Homework and Programming Assignments

There will be ordinary written homework assignments during the term, small programming assignments, and three larger-scale programming projects. Everything you turn in for grading must show your name, your computer account, your discussion section number, and your tutorial time. You will turn in all projects and homeworks electronically.

Testing and Grading

In addition to homework, there will be two midterms during the term, and a final. All tests are open book, open notes, and are cumulative.

Each of the two midterms will count for 25 points, and the final will be worth 50 points. The three programming projects count for 60 points. The homeworks count for 24 points, and there are 16 points to be distributed for tutorial participation at the discretion of your TA. So be nice to him/her. Your letter grade will be determined by total points out of the possible 200. The actual grade cutoffs will be set at some later time. I want to emphasize, however, that the class *will not* be curved.

If you believe we have misgraded an exam, return it to your TA with a note explaining your complaint. We will regrade the *entire* test. You should check the on-line solutions first to make sure that this regrade will make your total score go up. By University policy, final exams may *not* be regraded. You may look at them at times and places to be announced.

I will grant grades of Incomplete *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. Do *not* try to get an incomplete simply as a way to have more time to study or do a project; that is contrary to University policy.

Policy on Collaboration and Cheating

I strongly encourage you to help each other on homework assignments (although not, of course, on tests) by discussing the work *before* you do it. But you should complete and write up each homework yourself.

Team projects are normally turned in once, no matter how many people worked on them. Make sure, of course, that the names of all the team members appear on the project write-up. The experience of working in a team is one of the most valuable aspects of this course. A very frequent request from the companies that hire our graduates is to give them more experience of realistic team programming. As in the real world, we attempt to assign individual credit in proportion to your effort, but the overall success of the project is the main factor that determines the grades of all the team members. So you all share the responsibility for making sure the project is completed fully and on time. You all get to bask in the glory for an

excellent project, or wallow in the ignominy for a bad one. Your project partner must be chosen from students in your tutorial.

Copying and presenting another person's work under any other circumstances constitutes cheating. Electronically submitted programs are particularly easy to check for copying or trivial changes, and we will be doing that. Any incident of cheating will be reported to the CS chair and to the Office of Student Conduct. I realize that CS and EE programs at Berkeley are very intense, and that students are often under extraordinary pressure to make deadlines. But deadlines are a fact of life, and will persist after college. The trick is to get ahead of them. You can seek advice from the TAs and lab assistants early if you feel yourself getting behind in something. Knowing where and how to get advice on things you don't understand is a skill everyone needs to succeed in the real world.

Lateness

We will give no credit for written homework turned in after the deadline, so that we can make on-line solutions available promptly and so that you can discuss those solutions in your discussion sections. Please do not beg and plead for exceptions; an individual assignment is worth too few points to justify your groveling at the my feet (a comment that probably applies to individual test questions, as well). You can miss an assignment or two and still get your A+. If some personal crisis disrupts your schedule one week, don't waste your time and ours by trying to fake it; just be sure you do the next week's work on time.

On programming projects, we are a tad more lenient. If an assignment is N days late, I'll penalize it $10N$ percent.

Scheduled Discussion Sections and Labs

All labs will be in 275 Soda. The following table is the current list of discussion and lab sections with their TAs:

AG is Alice Gutman, BC is Billy Chen, CS is Chris Shaw, GM is Geoff MacDonald, JnS is John Sampson, and JfS is Jeff Schoner

Disc 101	W,F 1000–1100	405 Soda	AG	Lab 101	Tu,Th 900–1100
Disc 102	Tu,Th 1300–1400	320 Soda	CS	Lab 102	M,W 1230–1430
Disc 103	Tu,Th 1400–1500	320 Soda	JnS	Lab 103	M,W 1430–1630
Disc 104	W,F 900–1000	320 Soda	JnS	Lab 104	Tu,Th 1230–1430
Disc 105	Tu,Th 1700–1800	320 Soda	BC	Lab 105	M,W 1630–1830
Disc 106	Tu,Th 1800–1900	320 Soda	GM	Lab 106	M,W 1830–2030
Disc 107	Tu,Th 1500–1600	320 Soda	JfS	Lab 107	Tu,Th 1600–1800