CS 61B: Data Structures Summer 2005

Course Overview

Instructor:
George Porter
Office: 473 Soda Hall
Phone: (510) 642-8919
Email: gporter@cs.berkeley.edu

Teaching Assistants:
Sourav Chatterji, cs61b-ta@cory.eecs.berkeley.edu
Nick Neely, cs61b-tb@cory.eecs.berkeley.edu
Instructor and Tas: cs61b@cory.eecs.berkeley.edu

Webpage:
http://inst.eecs.berkeley.edu/~cs61b/su05/

Newsgroup:
ucb.class.cs61b

Class Information:

Location:	150 GSPP (across the street from Cory and Soda)
Lecture Times:	M, Tu, W, Th 11-12:30pm
CCN:	28935
Units:	4
Duration:	6/20/05 through 8/12/05

Course Overview:

Roughly a quarter of the semester will be devoted to an introduction to Java. Constructs and topics to be covered include the following:

- The compile/execute cycle.
- Primitive data types (integer, floating point, character, boolean); arrays; classes.
- Interactive control structures.
- Functions; recursion; overloading.
- Inheritance; interfaces; exceptions; threads.

In the rest of the semester, and in conjunction with practice of basic Java programming techniques, we will implement and experiment with fundamental algorithms and data structures:

- Construction, modification, and traversal of linked list structures of various forms -- singly-linked, doubly-linked, and circular, with and without sentinels.
- Construction, modification, and traversal of binary trees (in particular, binary search trees and expression trees).
- Sorting of sequences by selection, insertion, quicksort, merge sort; binary search through a binary search tree of a sorted sequence.
- Binary heaps.

- Hashing.
- Elementary graph structures and algorithms.

Your aim will be to recognize when these data structures and algorithms are applicable to a problem, and to be able to evaluate their relative advantages and disadvantages.

Design in terms of abstract data types and isolation of their implementation in modules will be emphasized. We intend that, having taken CS 61B, you will:

- understand the distinction between a specification or interface and an implementation;
- understand pre- and post-conditions in specifications;
- be able to use a specification expressed as a set of procedure headers with comments; and
- be able to provide suitable comments for modules, data types, and functions.

Data types used for illustration will include queues, stacks, dictionaries, sets, and GUI toolsets.

Discussion Sections

CCN	Day	Time	Room	TA
28940	TuTh	1-2pm	320 Soda	Neely
28950	TuTh	3-4pm	320 Soda	Chatterji

Lab Sections

CCN	Day	Time	Room
28945	MW	1-3pm	275 Soda
28955	MW	3-5pm	275 Soda

Lecture and Discussion section attendance is strongly encouraged. Lab attendance is required. Due to space limitations, you must attend the lab that you are registered with via Telebears.

Textbooks

	Author	Title	Publisher	ISBN
Required	Weiss	Data Structures & Problem Solving Using Java 3rd ed	Addison Wesley	0-321-32213-4
Optional	Sierra & Bates	Head First Java	O'Reilly	0-596-00465-6

Data Structures & Problem Solving Using Java is the primary textbook for this course, and you will need to have it. If you have a good working knowledge of Java and feel comfortable writing Java programs, you don't have to buy *Head First Java*.

Prerequisites

The prerequisite for this course is CS 61A. We expect to build heavily on data-oriented and object-oriented design approaches introduced in those courses, as well as on algorithms for recursive list and tree manipulation. If you have not taken this course, you will likely not be

able to take 61b. If you feel that you have taken a substantially similar course, then there are appeal forms in the main CS office (387 Soda hall).

If you have already taken a data structures course, and are interested in just learning Java, then you may not need to take 61b. You might be able to take CS 9G, which is a Java course. Additionally, if you have taken a simplified data structures course, or have advanced placement credit and the like, you may be able to take a self-paced alternate to 61b. If you feel like you already know the material in this course, please see me.

Grading Policy

Homework	10%
Labs	10%
Midterm I	15%
Midterm II	15%
Project I	8%
Project II	10%
Project III	12%
Final	20%
Total	**100%**

The exams (and of course the Final) are cumulative, meaning that material from earlier in the course may appear on any exam. In addition there are homework, lab, and project grades.

Homework problems are due the following Monday at the beginning of class. Please see the course schedule for homework due dates. Updates about due dates may also be posted to the newsgroup or announced in class.

Your lowest homework grade will be dropped, and your lowest two lab grades will be dropped. Homeworks and labs may not be turned in late. We drop your lowest homework grade and two lowest lab grades to accommodate conflicts, etc. Do not ask for further accommodation. Labs must be "checked off" in your assigned lab period.

Exams will be graded by the instructor and TAs.

Regrade policy: If you feel that an error was made in the grading of your exam, you can write up a case for why you think you should get a regrade. Staple this to the front of your exam and turn it back to us within one week of getting your exam back. Keep in mind that we may regrade your entire exam. Your resulting score may be higher, the same, *or lower* than the original. Proceed with caution. The final will not be regraded, according to University policy.

Project "slip days" policy: Because the summer semester moves so quickly, we cannot allow much extra time for turning in projects. You are allowed **3** slip days during the semester. **But you can only use at most two on one project!** A slip day is counted for each day you are late. A "day" is considered any amount of time up to 24 hours. So if you turn in the project 15 minutes late, that is one slip day. If you turn in the project 24 hours and 15 minutes late, that is two slip days. If you turn your project in 48 hours and 15 minutes late, it won't be graded and you will not receive credit for it.

Collaboration and Cheating

The University takes academic dishonesty very seriously. All incidences of cheating will be reported to the Office of Student Conduct. This includes exams, labs, projects, even homework. Additionally, if you cheat on an assignment you will receive the negative of the full score. This means that if you cheat on a project and it is worth 10 percent of your grade then you would get –10 points for that assignment. In keeping with policies from previous semesters of 61b, any student caught cheating on a project can receive an F in the course. Please don't go down this route!

We want you to be able to go to your classmates for help during the semester. Examples of **acceptable** help include:

- Explaining how to use the tools, including text editors, the java compiler, and the turn-in utility
- Explaining Java error messages and what they mean
- Discussing questions raised during lecture, discussing examples from the textbook.
- Talking about strategies for solving problems (say, on the homework), without sharing homework writeups or code. If you had input on a homework/lab/etc from another student, you must credit them in your writeup.

Examples of **unacceptable** collaboration include:

- Getting code from a classmate, the Internet, or previous semesters. (You may of course exchange project-related code with your project group).
- Giving your code to anyone, putting it on the Internet, on a web page, etc. (see project exception above)
- The above two rules are called the "No Code Rule"
- Working on homework writeups together and submitting them as your own. If another student gives you help on the structure of a homework, that is fine as long as you write up your own, independent solution and attribute your friend in the writeup.
- Presenting another students work as your own

For projects, you can share code within your group. Note that we will employ special software to detect cheating.

Please talk to the instructor or TAs if you have any questions about what is and what is not acceptable. The best way to avoid academic dishonesty is to stay ahead of the material, and seek out help from us when you start getting behind. The earlier you come to us the better the situation will be for you.

Workload

CS 61b is a heavy workload class, especially during the summer. We're going to be moving through 15 weeks of material in 8 weeks. My advice is to keep ahead of the work by spending time each day on the class. Specifically, I have marked the homework problems with the lectures they correspond to. Start working on them the day we cover that material in class (and at least look the problems over before class). Start working on the project the day it is handed out, and plan on finishing a few days early. This will allow slack for hard-to-fix bugs and errors.

Projects

There will be three projects during the course. The first will be done individually, and the second and third will be done in groups of two or three. Details on each of these projects will be provided as needed during the semester.

Schedule (subject to change)

Lecture #	Date	Topic
1	20-Jun	Introduction / OO intro
2	21-Jun	types, operators, conditionals, loops
3	22-Jun	references, strings, arrays
4	23-Jun	objects, classes, methods, packages, constructors
5	27-Jun	inner classes, using objects, java library
6	28-Jun	types, inheritance, polymorphism
7	29-Jun	interfaces and exceptions
8	30-Jun	Lists
9	5-Jul	Java Review
10	6-Jul	Midterm 1
11	7-Jul	In-class Case Study
12	11-Jul	Asymptotic analysis
13	12-Jul	Asymptotic analysis
14	13-Jul	Sorting
15	14-Jul	Sorting
16	18-Jul	Stacks & Queues
17	19-Jul	Hash tables
18	20-Jul	Hash tables
19	21-Jul	Midterm 2
20	25-Jul	Trees & Tree traversals
21	26-Jul	Binary search trees
22	27-Jul	Balanced trees
23	28-Jul	Paired programming
24	1-Aug	Priority queues
25	2-Aug	Graphs
26	3-Aug	Weighted graphs
27	4-Aug	Weighted graphs
28	8-Aug	Amortized analysis
29	9-Aug	Large-scale Programming Methodology
30	10-Aug	Final
31	11-Aug	Final

Students with special needs

Students with special needs for class or exams must obtain written notice from the University Disabled Students Program. Please provide this notice to the instructor during the first week of class so that necessary arrangements can be made.