Computer Science 051 – Spring 2009

Instructors & Text

Instructors: **Kim Bruce**
222 Edmunds, x7-1866
kim@cs.pomona.edu
Office Hours: MTWF 1:30-3:00 p.m., & by appt.
Yi Chen
223 Edmunds, x7-8651
tzuyi@cs.pomona.edu
Office Hours: MW 4 - 5 p.m., R 3-4 p.m., & by appt.

Lectures: *Lec 1*: MWF 11:00 - 11:50 a.m.,
Lec 2: MWF 10:00 - 10:50 a.m., both in Lincoln 1135

Labs: *Lab 1*: M 7 p.m. - 10 p.m.,
Lab 2: T 9 a.m. - 12 p.m., both in Edmunds 229

Text: *Java: An Eventful Approach* by Kim B. Bruce, Andrea Pohoreckyj Danyluk, and Thomas P. Murtagh, Prentice Hall, 2006.

Course web page: www.cs.pomona.edu/classes/cs051/

Instructors' web pages: Bruce: www.cs.pomona.edu/~kim/
Chen: www.cs.pomona.edu/~tzuyi/

Overview

Computer Science 051 is an introduction to algorithm development emphasizing object-oriented design. Algorithms will be implemented as programs in the Java programming language. We will introduce data structures and recursion as tools to construct correct, understandable, and efficient algorithms. These topics will be developed further in CSCI 052, CSCI 062, and more advanced computer science courses. For those who wish a good introduction to the science of computing. we highly recommend the combination of CSCI 051 and CSCI 052, which includes an introduction to a broad array of topics in computer science. For those who wish a more intensive introduction to programming, we suggest the combination of CSCI 051 and CSCI 062.

By the end of this course you should have a good basic understanding of how to develop (design, code, and debug) medium-sized programs in Java, and have a basic understanding of how one might analyze programs for correctness and efficiency.

This course is a prerequisite for all upper level Computer Science courses. In Computer Science 051 *we do not assume that you have had any previous computer programming experience.* If you have had extensive previous experience you might consider CSCI 052 or CSCI 062. Please discuss this with your instructor *immediately* if you feel you fall into this category.

Lectures and Readings

The schedule below shows the topics to be covered at each class meeting. Consult the on-line course syllabus (see URL above) regularly to see the most current version of the schedule.

All reading assignments are from the text. Students should come to class having completed the indicated readings for the day. You should attempt to work all the problems at the end of each section as you are reading. Chapter review problems will be assigned during each lecture and should be completed by the next lecture.

Lecture	Date	Topic	Reading	Hmwk assigned
1.	Jan. 21	Fundamentals & Graphics	Ch 1	1.6.2, 1.9.3
2.	Jan. 23	Event Handling & Numbers	Ch 2, 3	2.7.2, 2.7.7
3.	Jan. 26	Conditionals	Ch 3, 4	3.11.2, 3.11.3
4.	Jan. 28	More conditionals	Ch 4	4.8.4, 4.8.6, 4.8.8
5.	Jan. 30	Primitive types & classes	Ch 5	5.7.2, 5.7.6
6.	Feb. 2	Class declarations	Ch 6	6.8.4, 6.9.1
7.	Feb. 4	Control Structures	Ch 6, 8	6.6.2, 7.2.1, 7.10.1
8.	Feb. 6	More Control Structures & Scope	Ch 7	7.11.3, 8.6.2, 8.6.3
9.	Feb. 9	Active Objects	Ch 9.1-9.2	9.10.1, 9.10.2
10.	Feb. 11	More Active Objects	Ch 9.3-9.5	9.10.3, 9.11.1
11.	Feb. 13	Even More active objects	Ch 9.6-9.9	9.11.2
12.	Feb. 16	Interfaces	Ch 10	10.5.1, 10.5.2
13.	Feb. 18	Design	Ch 21	21.9.1-21.9.4
14.	Feb. 20	GUI components	Ch 11.1-11.3	
15.	Feb. 23	More GUI components	Ch 11.4-11.6	11.9.1
16.	Feb. 25	More GUI components	Ch 11.7-11.8	11.9.2
17.	Feb. 27	Inheritance	Ch 17.1-17.5	
18.	Mar. 2	More Inheritance	Ch 17.6-17.7	17.8.5-17.8.7
19.	Mar. 4	Hand out test prog & review		

Lecture	Date	Topic	Reading	
20.	Mar. 6	*Midterm*		
21.	Mar. 9	Recursion	Ch 12.1	
22.	Mar. 11	More recursion	Ch 12.1	12.1.2, 12.1.4
23.	Mar. 13	Even more recursion	Ch 12.2	12.1.10
	Mar. 16-20	**Spring Break**		
24.	Mar. 23	For loops	Ch 13	13.6.2, 13.6.5
25.	Mar. 25	Arrays	Ch 14.1-14.5	14.2.2, 14.9.3
	Mar. 27	**Holiday - Chavez Day**		
26.	Mar. 30	More Arrays	Ch 14.6-14.8	14.9.4
27.	April 1	Even More Arrays	Ch 14.6-14.8	
28.	April 3	Multi-dimensional Arrays	Ch 15.1-15.2.2	15.4.1
29.	April 6	More multi-dimensional Arrays	Ch 15.2.3-15.3	15.4.2
30.	April 8	More multi-dimensional Arrays	Ch 15.2.3-15.3	15.4.4
31.	April 10	Strings	Ch 16.1-16.2	16.5.2-16.5.4
32.	April 13	More Strings	Ch 16.3	16.5.5, 16.5.6
33.	April 15	Even More Strings	Ch 16.4	16.5.7
34.	April 17	Exceptions	Ch 18	18.5.1, 18.9.1
35.	April 20	Streams	Ch 19.1-19.3	19.7.2, 19.7.3
36.	April 22	More Streams & Networks	Ch 19.4	19.7.4
37.	April 24	More Networking	Ch 19.5-19.6	19.7.5
38.	April 27	Searching	Ch 20.1	20.6.1
39.	April 29	Sorting	Ch 20.2-20.3.4	20.6.3
40.	May 1	More Sorting	Ch 20.3.5-20.5	20.3.8
41.	May 4	Concurrency		
42.	May 6	Summary		

Programming Assignments and Laboratories

There will be two types of programming assignments: *laboratory programs* that, as a unit, count for 15% of your final grade, and two *test programs* that make up 40% of the final grade. Test programs are treated as take-home exams, with no assistance allowed. All programs assigned during the semester should be completed following the guidelines in the *Academic Honesty Policy* described below.

Laboratory programs will not be accepted after the due date, as it is essential to keep up to date so that you will be ready for the next week's assignment. Late test programs will be accepted, but with a penalty of 10% per day.

Labs for this course will be held on Monday evenings from 7 to 10 p.m or Tuesday mornings from 9 a.m. to noon in 229 Edmunds. The room is equipped with iMac computers. We will be using the free Eclipse IDE (Interactive Development Environment) to create and run all of the programs in the course (sample programs, lab exercises, and homework assignments).

The purpose of the lab sessions is to provide a time during which your instructor can actively assist you in the development of *laboratory programs*. Attendance at these lab sessions is **mandatory**. The initial lab period will familiarize you with the use of the course software. Remaining labs will be handed out in advance, and you will be expected to have planned your approach to the assignment before the beginning of lab. The lab session will then be used to develop the program and perform related exercises. You will submit your laboratory programs electronically. The procedure will be explained in laboratory. *Laboratory programs will generally be due by 11:00 P.M. two days after your lab session.*

Monday/Tuesday	Laboratory Title
Jan. 26/27	Intro. to Java
Feb. 2/3	Event Handling with Conditionals
Feb. 9/10	Defining Classes
Feb. 16/17	Parameters
Feb. 23/24	Loops and Simple Threads
March 2/3	GUI Practice
Wednesday, March 4	**Test Programs 1 distributed**
Friday, March 6	**Midterm Exam in class**
March 9/10	**Work on test program**
Friday, March 13	**Test Programs 1 Due at 4:00 P.M.**
March 23/24	Recursion I
March 30/April 1	Recursion II
April 6/7	Arrays
April 13/14	Two-D Arrays
April 20/21	Strings
Wednesday, April 22	**Test Program 2 distributed**
April 27/28	Streams
Wednesday, April 29	**Test Program 2 design due**
May 4/5	Work on test program
May 6	**Test Program 2 Due, 4:00 P.M.**

Exams

There will be an in-class midterm (worth 20% of your grade). There will also be a scheduled final exam (counting 25%).

- *Midterm examination*: Friday, March 6, in class. A sample exam will be made available later.
- *Final examination*: For both sections: Tuesday, May 12, at 2 p.m. *This is the time for shared exams – please check early on and let us know if you have a conflicting exam!* A sample exam will be made available later.

Grading Summary

Programs:			55%
	Laboratory Programs	15%	
	Test Programs	40%	
Exams:			45%
	Midterm Exam:	20%	
	Final Exam:	25%	
Total:			100%

Collaboration & Academic Honesty Policy

We highly encourage students to get together in small groups to go over material from the lectures and text, work problems from the text, study for exams, and to discuss the general ideas and approaches to laboratory assignments. However, work to be turned in, including programming assignments, must be done independently, unless we explicitly designate an assignment as one in which collaboration is allowed. As expained in the student handbook, this means that the work you turn in must represent only your own work. It must not be based on help from others or information obtained from sources other than those approved by the instructors (e.g., the text, web pages linked from the course web page, and materials provided in lecture). Effective learning is compromised when this is not the case.

Accordingly, you should never read or copy another student's code or solutions, exchange computer files, or share your code or solutions with anyone else in the class until after the assignment is due. Under no circumstances may you hand in work done by, or in collaboration with, someone else under your own name, with the exception that you may freely use any code that we provide to you or code from the textbook, as long as you cite this code as coming from the instructors or the book. Additionally, the student mentors are allowed to help you with your code.

When a program is assigned, we will identify it as either a "test" or a "laboratory"' program. The academic honesty policy applies differently to each with respect to collaboration or assistance from anyone other than the mentors or instructors:

Test Programs. Any assignment designated as a test program is to be treated exactly as a take-home, open-book test with respect to rules for obtaining assistance. You are allowed to read your textbook, class notes, and any other source approved by your instructors. You may not consult anyone other than your instructors. The instructors encourage the asking of questions, but reserve the right not to answer, just as you would expect during an exam. Guideline: Any work that is not your own is considered a violation of the academic honesty policy.

Laboratory Programs. Laboratory programs are expected to be the work of the individual student, designed and coded by him or her alone. Help locating errors is allowed, but a student may only receive help in correcting errors of syntax; help in correcting errors of logic is strictly forbidden. Guideline: Assistance from anyone other than the mentors or instructors in the design or coding of program logic will be considered a violation of the academic honesty policy. The only time it is acceptable to look at another student's code is to help them identify an error in their program. You may not provide them with suggested code to correct it nor may you copy their code in your program. You may point out similar examples from the text or lecture notes that help them correct the error.

If you do not understand how the academic honesty policy applies to a particular assignment, consult with us. When in doubt, credit the people or sources from whom you got help. This also goes for any help obtained via the Internet. You will not lose any points for acknowledging help obtained where the rules for assistance are unclear. If you are ever unsure about what constitutes acceptable collaboration, just ask.

Failure to abide by these rules is considered plagiarism, and will result in severe penalties. Violations are easy to identify and will be dealt with promptly. The first offense typically results in failure in the course. A second offense is automatically referred to the College's Board of Academic Discipline. See the Academic Honesty Policy in the Student Handbook for further information. Please do not put us, yourself, or anyone else in this unpleasant situation.