

CSE 8B: Introduction to Computer Science: Java

Spring 2009

Course Basics

(additional IMPORTANT information is on the class web page. READ IT!)

Instructor: Dr. Beth Simon (call me “Beth” or “Dr. Simon”)
Office: CSE 4104
Phone: 858-534-5419
Email: bsimon@cs.ucsd.edu (but use WebBoard or email TA for faster responses!)

Office Hours: Tuesday 11-12:30pm in CSE 4104
and by appointment

TA (Lisa Cowan) Office Hours TBD (see web page)

Class Meetings: T TH 9:30-10:50AM Center 214

Discussion Section: Monday 4-4:50 Center 214 Attendance highly recommended

Open Lab Tutoring: Available in CSE basement labs as indicated on the class web page. These tutors exist to help you learn and to provide a second set of eyes for debugging. They will not tell you what code to write. See information on the web site on “How to prepare to ask a question”.

Text: Required (2):
Mark Guzdial and Barbara Ericson, Introduction to computing and programming with Java: A Multimedia Approach, April 2006. Prentice Hall.
Savitch, Walter. Absolute Java, **8B Special Printing (available at bookstore) OR Second Edition or Third Edition**, Prentice-Hall.

Problem Solving Assignments (PSAs):

Every 2 weeks there will be a set of problem solving assignments due. You should plan on spending 10-12 hours – IF you spread the work out over multiple days. If you try to do it all in one sitting, you should expect to perform poorly.

Code should be developed on ieng6.ucsd.edu and submitted with the **bundle script in the lab linux environment** by the deadline. **ALL CODE MUST BE EXECUTABLE WITH NO ERRORS ON THE ieng6.ucsd.edu ENVIRONMENT.**

Quizzes: There will be quizzes every other week on Tuesdays (weeks 3,5,8,10). Alternate weeks are required PeerWise submissions (weeks 2, 4, 6, 7, and 9). (See details on web).

Exams: There will be one midterm and one final exam. They will cover all the assigned reading, lecture material, and assignments up to the class period BEFORE the exam (eg, the exam is on Tuesday, material up through the previous Tuesday is fair game). **Exams must be taken at the scheduled time.**

All quizzes and exams are closed book, closed notes, no calculators allowed.

The midterm exam will be on Tuesday May 5 (Week 6), during the normal class time.

The final exam will be on Tuesday, June 9 from 8:00 AM to 10:59 AM.

Grading:

Your grade will be based on the weighted average of your scores on PSAs, quizzes, exams, clicker participation and other critical required activities (things absolutely critical to your learning professional development, and our support of your learning, but we need you to do outside class).

The final grade will be weighted as follows:

Critical Required Activities	2% (complete ALL for full 2%)
In-Class Quizzes	9%
PeerWise Quizzes	4%
iClicker Participation	2% (for clicking in 80% of time)
Midterm	23%
PSAs	30%
Final	30%

I will grade on a 90%, 80%, 70%, 60% scale. “Plusses” and “minuses” are given out only at my discretion. ***All grades are FIXED within 7 days of the grade being posted on gradesource -- no regrade requests will be taken after that time.***

ADDITIONALLY, YOU MUST SCORE AT LEAST A 55% ON THE FINAL EXAM TO PASS THE COURSE. ANYONE WHO SCORES LESS THAN A 55% ON THE FINAL EXAM WILL RECEIVE AN F – REGARDLESS OF THE PERCENTAGES DESCRIBED ABOVE.

Academic Integrity:

I and the class staff will be working hard to help you learn both the required concepts and gain the skills necessary to be able to solve problems using a computer. We expect your help in achieving these goals. If you cheat, you not only fail yourself (in that you won't be able to solve problems asked of you by employers), but you fail UCSD (by lowering the expected quality of our undergraduate degrees). You don't want either of those. So don't cheat.

In this class, you will be completing PSAs by programming by yourself – NOT IN PAIRS. While it is true that much real world programming happens in teams, there are still expectations of an individual programmer to complete (independently) subparts of a larger project. You will be developing that proficiency in this and many later classes. When you run into problems, barriers, or bugs -- you should attempt to work through them on your own first. There are a variety of techniques we recommend including:

- Identify the location of the error (either from the compiler error, or through use of a debugger or print statements for runtime errors).
- Read and trace (manually your code).
- Read in the textbook or look at Java API method descriptions on the web.
- Review lecture notes.
- Put in print statements printing appropriate variable values.
- Take a break (get a drink, sleep).
- Try a different input value.
- Comment out complicating code.
- Ask a tutor.

We repeat, it is important that you try to debug and fix your errors on your own. However, we also know (and you know) that sometimes it is a lot easier for a fresh pair of eyes to "see" a bug. Can you ask another student in the class? Maybe. See the Integrity statement online and the results of your classmates' decisions as posted on the class web site.