University of Washington
Computer Science & Engineering 142: Introduction to Programming I, Autumn 2007
Course Syllabus

Instructor

name: Marty Stepp
email: stepp@cs.washington.edu
office: CSE 466
office phone: (206) 685-2181
office hours: *see course web site*
lecture time: A: MWF 9:30 AM - 10:20 AM, KNE 210
B: MWF 11:30 AM - 12:20 PM, KNE 120

Course Administrator

name: Pim Lustig
email: pl@cs.washington.edu
office: CSE 126
office phone: (206) 616-3225

Pim will handle many course details including registration and switching sections.

Course Overview

This course provides an introduction to computer science using the Java programming language. CSE 142 is primarily a programming course that focuses on common computational problem solving techniques. No prior programming experience is assumed, although students should know the basics of using a computer (e.g., using a web browser and word processing program) and should be competent with math through Algebra 1. Students with significant prior programming experience should consider skipping CSE 142 and taking CSE 143 (we allow students to do so without any special permission).

Discussion Sections

You will be expected to participate in a weekly discussion section, held on various times and places on Thursdays (see the course web site for details). The TA who runs your section will grade your homework assignments. In section we will answer questions, go over common errors in homework solutions, and discuss sample problems in more detail than we can in lecture.

Each student will be assigned a section participation score that is weighted the same as one homework assignment. You will receive **3 points** for each section you participate in, up to a maximum of 20 points.

Course Web Site

http://www.cs.washington.edu/142/

You should check the course web site daily for any important course-related announcements.

Textbook

Reges/Stepp, *Building Java Programs: A Back to Basics Approach*. ISBN 0321382838. Required.

No assignments or required readings will be given directly from the textbook, and you may choose not to purchase it if you wish. However, the book was written specifically for this course and makes a useful supplement to the lecture presentations. Also, exams in this course will be open-book, so it may be advantageous to own the book for use as a reference during exams.

Computer Access and Software

The department operates an Introductory Programming Lab (IPL) located in room 334 of Mary Gates Hall. TAs and consultants will be available at the lab to help students with problems. The recommended software for the course is the Java Development Kit (JDK) version 6 and the DrJava editor.

The course web site contains links to download this software free of charge if you wish to work at home.

Grading

Your grade percentage in the course comes from completing the following tasks:

50% weekly homework assignments (including section participation)
20% midterm (**Friday, November 2, 2007**, *in class*)
30% final exam (**Wednesday, December 12, 2007**, *time/place to be announced on course web site*)

Your percentage is mapped onto the 4.0 grade scale roughly as follows. You are guaranteed at least the grade shown below for the percentage shown.

90%: at least 3.5	85%: at least 3.0	80%: at least 2.5
75%: at least 2.0	70%: at least 1.5	60%: at least 0.7

Exams

The exams in this course are open-book and open-notes. You may bring any written materials you like, such as textbooks, printed handouts, homework assignments, or other programs. No electronic devices may be used. Generally, make-up exams will not be given without substantial extenuating circumstances. If you need to miss an exam, you must contact your instructor by phone or email and receive permission to do so *prior* to the exam. Even if you are sick, you should email or call your instructor to leave a message that you need to be contacted.

Homework

Homework in this course will consist of weekly programming assignments to be completed individually and submitted electronically from the course web site. Most programs will be graded on a 20-point scale based on "external correctness" (behavior) and "internal correctness" (the style in which the program is written). Disputes about homework grading accuracy must be made within 2 weeks of the assignment's due date.

Lateness Policy

Each student receives **5 free "late days"** for use on homework assignments. A late day allows you to submit a program up to 24 hours late without penalty. For example, you could use 2 late days and submit a program due on Wednesday on Friday with no penalty. Once a student has used up all of their late days, each successive day that an assignment is late will result in a loss of 1 point on that assignment. Regardless of how many late days you have, you may not submit a program more than **4 days** after it is due or after the last day of class. Students will not be given extensions unless they have highly extenuating circumstances as decided by the instructor.

Academic Integrity and Collaboration Policy

Programming assignments must be completed individually. You may discuss an assignment in general terms with other students, including a general discussion of how to approach the problem, but all code you submit must be your own. Any help you receive from classmates should be limited and should never involve details of how to code a solution. You must abide by the following:

- You may not work as a partner with another student on an assignment.
- You may not show another student your solution to an assignment, nor look at his/her solution.
- You may not have another person (current or former student, tutor, friend, TA, etc.) "walk you through" an assignment (such as by describing in detail how to solve it, or sitting with you as you write it).

You are also responsible for taking reasonable means to ensure that your work is not copied by others. This includes making sure to log out or lock any shared computers.

If you are retaking the course, you are allowed to submit a previous solution that you authored unless that program was involved in a case of academic misconduct. For any assignment where academic misconduct was found (whether the case was settled formally or informally), you must write a new version of the program.

We will enforce this policy by periodically running similarity detection software over all submitted programs. Be aware that over **60 of 420 students** were charged with violating this policy in Autumn 2006. Many cases led to reduced grades and/or marks on permanent records.