

University of Washington
Computer Science & Engineering 142: Introduction to Programming I, Winter 2008
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, GUG 220
B: MWF 11:30 AM - 12:20 PM, GUG 220

Course Administrator

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

Pim handles 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, so you may choose not to purchase it if you like. 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 jGRASP editor.

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

Grading

| | |
|-----|--|
| 50% | weekly homework assignments (including section participation) |
| 20% | midterm (Friday, February 15, 2008, in class) |
| 30% | final exam (Wednesday, March 19, 2008, time/place to be announced on course web site) |

This maps to the 4.0 scale roughly as follows. You will get at least the grade 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

Our exams are open-book and open-notes. You may bring any written materials, such as textbooks, printed handouts, homework assignments, or programs. No electronic devices may be used. Make-up exams will not be given except in the case of a serious emergency. If you need to miss an exam, phone or email the instructor and get permission *before* the exam. Even if you are sick, you should contact the instructor to leave a message.

Homework

Homework consists of weekly programming assignments done individually and submitted electronically from the course web site. Programs will be graded on "external correctness" (behavior) and "internal correctness" (style and design). Disputes about homework grading must be made within 2 weeks of receiving the grade.

Lateness

Each student receives **5 "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 Tuesday on Thursday 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 extenuating circumstances as decided by the instructor.

Academic Integrity

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 or provide to 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 "walk you through" an assignment, describe in detail how to solve it, or sit with you as you write it. You also may not provide such help to another student. This includes current or former students, tutors, friends, TAs, web site forums, or anyone else.

Under our policy, a student who gives inappropriate help is equally guilty with one who receives it. You must not share your solution code and ideas with others. You must also take reasonable means to ensure that your work is not copied by others. This includes making sure to log out or lock shared computers, not leaving printouts of your code in public places, and not emailing your code to other students or posting it on the web.

If you are retaking the course, you may resubmit a previous solution unless that program was involved in an academic misconduct case. If misconduct was found, you must write a new version of that program.

We enforce this policy by running similarity detection software a few times per quarter over all submitted student programs, including programs from past quarters. Students who violate the policy are given reduced scores and sometimes sent to a University committee. This can lead to marks on permanent academic records.

We take this policy very seriously. **54 of 530 students** were charged with violating this policy in Autumn 2007. Please contact the instructor if you are unsure whether a particular behavior falls within our policy.