

University of Washington

Computer Science & Engineering 143: Introduction to Programming II

Course Syllabus, Winter 2009

Instructor

name: Marty Stepp
email: stepp@cs.washington.edu
office: CSE 466
office phone: (206) 685-2181
office hours: *see course web site*

Course Administrator

Pim Lustig
pl@cs.washington.edu
CSE 126
(206) 616-3225
Pim handles many course details, e.g. registration and switching sections.

Course Overview

This course is a continuation of CSE 142. While CSE 142 focused on control issues (loops, conditionals, methods, parameter passing, etc), CSE143 focuses on data issues. Topics include: ADTs (abstract data types), stacks, queues, linked lists, binary trees, recursion, interfaces, inheritance, and encapsulation. The course also introduces the notion of complexity and performance tradeoffs in examining classic algorithms such as sorting and searching and classic data structures such as lists, sets, and maps. The course will include a mixture of data structure implementation and using off-the-shelf components from the Java Collections Framework. The prerequisite is CSE 142 or equivalent.

Lecture Time

MWF 3:30 PM - 4:20 PM, KANE 120

Discussion Sections

You will be expected to participate in two weekly 50-minute discussion sections, held on various times and places on Tuesdays and 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.

Course Web Site

<http://www.cs.washington.edu/143/>

All resources from class will be posted here. Check the web site daily for any important course-related announcements.

Textbook

Reges/Stepp, *Building Java Programs: A Back to Basics Approach*. ISBN 0536240167 (or 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. It contains practice problems and online videos you can use to study for your exams. Also, the exams in this course will be open-book, so it may be advantageous to own the book to bring 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 Eclipse and jGRASP editors, but you may use any editor you like.

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

Grading

| | |
|-----|---|
| 50% | weekly homework assignments |
| 20% | midterm (Wednesday, February 18, 2009, in class) |
| 30% | final exam (Thursday, March 19, 2009, time/place TBA) |

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 | 80%: at least 2.5 | 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, including calculators.

Make-up exams will not be given except in case of a serious emergency. If you must miss an exam, even if you are sick or injured, you must contact the instructor *before* the exam (or arrange for someone to do so). You must show evidence that you are physically unable to take the exam, such as a doctor's note specifically mentioning the CSE 143 midterm, before the exam. No make-ups will be granted for personal reasons such as travel, personal hardship, to ease your exam week schedule, or leisure. No student will be permitted to take an exam early for any reason.

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 Thursday 11pm on Saturday by 11pm with no penalty. Once a student has used up all the 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 on homework assignments unless they suffer extenuating circumstances or emergency as decided by the instructor.

Academic Integrity

Programming assignments must be completed individually; all code you submit must be your own work. You may discuss general ideas of how to approach an assignment, but never specific details about the code to write. 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. Instead of providing such help to someone who does not understand an assignment, point them to other class resources such as lecture examples, the textbook, the IPL, or emailing a TA or instructor. You must not share your solution and ideas with others. You must also ensure that your work is not copied by others, such as making sure to log out of 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. Please contact the instructor if you are unsure whether a particular behavior falls within our policy.