

University of Washington

Computer Science & Engineering 143: Introduction to Programming II

Course Syllabus, Winter 2011

Instructor

name: Marty Stepp
email: stepp@cs.washington.edu
office: CSE 636
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 all issues of 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), CSE 143 focuses on data issues. Topics include: abstract data types (ADTs), lists, stacks, queues, linked lists, binary trees, recursion, interfaces, inheritance, and encapsulation. The course also introduces the notion of complexity and performance trade-offs 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 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 answer questions, go over common errors in homework solutions, and discuss sample problems in more detail than we can in lecture.

Short "**pre-section problems**" will be assigned on the course web site that will be due in section, worth a small number of points. You must attend section to turn these in by hand; they can not be made up, turned in late, or submitted online. You will need to submit at least half of these problems (one of the two problem sets each week) to receive full credit.

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, 2nd edition*. ISBN 0136091814. **Required.**

"Pre-section problems" and lecture readings will be assigned directly from the textbook, so you are required to either own a copy or have frequent access to one. (Weekly homework assignments have their own spec documents on our web site, not directly from the textbook.) The book was written specifically for this course and makes a useful supplement to the lecture presentations; it also contains practice problems and online videos you can use to study for your exams. 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.

The book can be bought in paperback or bound (cheaper) editions from the bookstore; either edition is fine for this course. The book can also be bought as a **digital download**, but you won't be able to use the digital version on your exams, so it might be less useful than the physical textbook. Older versions of the book, such as the 1st edition or "preliminary custom editions", don't have all of the problems/chapters we will assign, so they are not satisfactory for use in this course.

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 (Friday, February 11, 2011, 3:30 - 4:20pm, Kane 120)
30%	final exam (Thursday, March 17, 2011, 2:30 - 4:20pm, Kane 120)

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** for the *Building Java Programs* textbook, but **closed-notes** and closed for all other resources. You may not bring or use any other books or printed materials such as handouts, slides, or practice exams. (A "cheat sheet" of necessary syntax will be given to you at the exam.) 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 else to do so). You must show evidence that you are physically unable to take the exam, such as a clear and specific doctor's note mentioning the date, exam, and reason. No make-ups will be granted for personal reasons such as travel, personal hardship, leisure, or to ease exam week schedules. No make-ups will be granted to students who contact us after the exam is over except in cases of dire emergency. No special accommodations will be made for students who arrive late to exams, regardless of the reason (missing a bus; overslept; sick; etc.). 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 **4 "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 3 days after it is due** or after the last day of class. Students will not be given extensions on homework assignments unless they have 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 **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, for any reason.
- 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.
- You **may not post your homework solution code publicly online**, such as to ask others for help. This includes public message boards, forums, file sharing sites and services, or any other online system.

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 periodically over all submitted student programs, including programs from past quarters. Students who violate the policy are offered reduced scores and sometimes sent to a University committee. This can lead to marks on permanent academic records. Please be careful, and contact the instructor if you are unsure whether a particular behavior falls within our policy.