University of Washington
Computer Science & Engineering 373: Data Structures and Algorithms
Course Syllabus, Winter 2013

Instructor

Marty Stepp
stepp@cs.washington.edu
Office: CSE 636; (206) 685-2181 (see course website for office hours)
Pim Lustig (pl@cs.washington.edu) handles all registration issues (adding, changing sections, waiting list, etc.).

Course Overview

This course is about fundamental data structures and algorithms. In this course, you will:

- Learn about fundamental data structures (including lists, stacks, queues, trees, sets, maps, heaps, and graphs)
- Learn about searching and sorting algorithms
- Understanding abstract data types (ADTs) and the tradeoffs between different implementations of ADTs
- Learn to use existing data structure libraries such as Google's Guava
- Gain an understanding of which data structures are most effective for various scenarios and problems
- Become proficient with analyzing the running time of various algorithms associated with data structures
- Implement several data structures in detail
- Gain familiarity with a computer's memory hierarchy and related structures such as B-trees

Prerequisite

CSE 143 or equivalent

Lecture Time

MWF 2:30 PM - 3:20 PM, EEB 105

Course Web Site

http://www.cs.washington.edu/373/

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

Textbook

Weiss, Mark A. *Data Structures and Algorithm Analysis in Java, 3ed,* Addison Wesley: 2011, ISBN 0132576279.

The book is listed as being required for the course. But problems and assignments will not be assigned out of the book. Despite this, it may be useful. The book describes lecture topics in more detail; most lectures will contain suggested book reading to help understand the topic. Also, exams in this course will be open-book, so it will be advantageous to own the book for use as a reference during exams.

Computer Access and Software

We will use Java 1.6 for programming assignments. We recommend (but do not require) that you use the Eclipse development environment. The College of Arts & Sciences Instructional Computing Lab (rooms B022 and B027 in the Communications Building) is the computer lab for this course; the above software should be installed there as well as in other public campus labs. Links for downloading and installing Java and Eclipse can be found on our course web page.

Grading

60%	assignments (programming projects and written exercises)	
20%	midterm	(**Wednesday, February 20, 2013**, *in class*)
20%	final exam	(**Tuesday, March 19, 2013**, 2:30 - 4:20 PM, EEB 105)

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, but closed-notes. 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 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 student will be permitted to take an exam early for any reason.

Assignments

Assignments will be a mix of programming projects and written exercises. Written exercises will be due at the beginning of class the date the assignment is due. Programming assignments will be submitted electronically from the course web site. Programming assignments 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

For full credit, written assignments must be turned in on paper **at the start of** lecture on the day it is due. You may turn in written assignments up to the start of the following lecture for a 10% deduction (i.e. 90% credit). Written assignments **will not be accepted** any later than the start of the lecture following the lecture the assignment was due. This policy includes written portions of programming assignments. Only in extenuating circumstances (e.g. sick, snow day) with the instructor's prior approval may you turn in a written assignment electronically (e.g. through email).

All programming assignments are due electronically at a specific time. Refer to the course calendar for due dates. Each student in the class will have a total of **three "late days"** (a late day is 24 hours of lateness). There are no partial days, so assignments are either on time, 1 day late, 2 days late, etc. Regardless of how many late days you have, **you may not submit a program more than 2 days after it is due**. Once a student has used up all of his or her late days, each successive late day will result in a loss of 10% on the assignment.

Academic Integrity and Collaboration

All assignments must be completed individually; all code and written exercise solutions 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 or solution 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 or specifically how to answer a question. You must abide by the following rules:

- 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, paid consultants, people on the Internet, or anyone else.
- You may not post your homework solutions online to ask others for help. This includes public message boards, forums, file sharing sites and services, or any other online system.
- You may not pay or otherwise recruit another person to complete some or all of an assignment for you, nor to give you extensive feedback on work you have done. This includes Rent-a-Coder, Craigslist, online forums, etc.

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, office hours, or emailing a TA or instructor. You must not share your solution 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 vigorously 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 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.