



CSE 373

Data Structures & Algorithms

Autumn 2010

Instructor: Ruth Anderson
Email: rea at cs.washington.edu
Office Location: Allen Center (CSE) 360
Office Hours: M 1-2pm, Thurs 11am-12pm or by appointment

Teaching Assistants: (Office Hours time and location TBA)

Timothy Jang, (jangt13 at cs.washington.edu)
Danushen Gnanapragasam, (nanabyte at cs.washington.edu)
Christopher Furmanczyk, (furmacc at cs.washington.edu)
Amanda Shen, (ashen at cs.washington.edu)

Lecture: MWF 2:30-3:20, SAV 264

Course Web Page: <http://www.cs.washington.edu/373/>

Overview and Goals: Achieve an understanding of fundamental data structures and algorithms and the tradeoffs between different implementations of these abstractions. Theoretical analysis, implementation, and application. Lists, stacks, queues, heaps, dictionaries, maps, hashing, trees and balanced trees, sets, and graphs. Searching and sorting algorithms.

Prerequisite: CSE 143

Course Text: Weiss, Mark Allen. **Data Structures and Algorithm Analysis in Java** 2nd Ed., Addison Wesley: 2007, ISBN: 0-321-37013-9

Assignments: Assignments will be a mix of written exercises and programming projects. Programming assignments will generally be due Thursday evenings and will be submitted electronically via the web. Exact dates and deadlines will be specified on each assignment.

Exams: We will have two midterm exams and a final exam. The final exam is scheduled for 2:30-4:20pm Tuesday, December 14, 2010. Makeup exams will only be offered under extraordinary circumstances; you should plan to attend the exams when they are given. Exams will normally be closed-book, closed-notes, and calculators will not be allowed.

Late Policy: Late work will be accepted but at a penalty of 20% off per 24 hours late. Note that ALL parts of the assignment must be received by the stated deadline. In the case of written assignments that are due at 2:30pm on Friday, you would need to create an electronic version and email it to us in order to submit by 2:30pm on Saturday to be considered 24 hours late. If unusual circumstances that are truly beyond your control prevent you from submitting an assignment or attending an exam on time, you should discuss this with the instructor, preferably in advance. (Even if you're sick in bed at home, you should still be able to make a phone call or send an email.)

Grading and Evaluation: Grades will be computed *approximately* as follows (weights may be modified):

- 50% - Assignments (Written Exercises and Programming Projects)
- 15% - Midterm Exam 1
- 15% - Midterm Exam 2
- 20% - Final Exam

Academic Integrity: Unless otherwise specified, you are to complete assignments individually. You may discuss the assignment in general terms (see description of Gilligan's Island rule on course web page and first day slides), but the code you write must be your own. You are encouraged to discuss ideas, approaches, concepts, bugs, etc., in English, but you may not show or give your code to anyone except this course's TAs and instructor. You are not allowed to write code with another student on an assignment or to show another student your solution to an assignment. Referring to solutions from this or other courses from previous quarters is also considered cheating.

Communications: The course message board is a good medium for discussing the course, getting help on assignments, and staying in touch outside of class hours. You can also email the instructor or TAs or go to office hours. In addition, the course staff will occasionally post announcements to the course email list. You will be automatically subscribed to the course email list if you are registered for the course and will be held responsible for anything posted there.

Computing Resources: We will use Java 6 for programming assignments. We recommend although will not require that you use the Eclipse development environment. The College of Arts & Sciences Instructional Computing Lab is the designated lab for this course; they have the above software installed, but the software should also be available in public campus labs. Links for downloading and installing Java and Eclipse can be found on our course web page.

Topic List:

- Review: data structure concepts, arrays, simple linked lists, different implementations of lists, stacks and queues, binary trees
- Introduction to complexity: O-notation
- Balanced search trees
- Heaps, priority queues
- Sets, including union/find algorithms
- Sorting and searching
- Dictionaries/maps, hashing
- Graphs

CS 373 – First Day Assignments

0) **Review Java and explore Eclipse** – These are available in the College of Arts & Sciences Instructional Computing Lab and other labs on campus. Instructions for installing on your home machine are on our course web page. Now would be a good time to review material from CSE 143!

1) **Assignment #1** – Your first programming assignment will be posted later today. Look for an email announcing its arrival to verify you are on the course mailing list!

2) **Preliminary Survey:** Please fill out the preliminary survey posted on our course web page by the evening of Thursday Sept 30th. (Course home page = <http://www.cs.washington.edu/373/>)

3) **Information Sheet:** Please bring a sheet of paper with the following information with you to lecture on Friday Oct 1st.

A Picture of you!
Student ID is o.k. but something more
interesting or readable is nice too.

Name (and what you like to be called)
Email address
Year (1,2,3,4 i.e. freshman, sophomore, grad, etc.)
Major
Hometown
Interesting Fact about yourself and/or what you did over summer break.

4) **Reading** in *Data Structures and Algorithm Analysis in Java*, 2nd Ed., 2007, by Weiss

- For this week:
 - › Chapter 1 – (review) Mathematics and Java (pp. 1-25)
 - › Chapter 3 – (Assignment #1) Lists, Stacks, & Queues
 - Lists (pp. 57-81, heavy on Java, much of this should be review)
 - Stacks (pp. 82-83)
 - Applications of Stacks (pp. 83-91, sections on “Postfix Expressions” and “Infix to Postfix Conversion” can be skipped, but read “Method Calls”)
 - Queues (pp. 91-95)
 - › Chapter 2 – (Topic for Friday) Algorithm Analysis (pp. 29-50)