CSE 332: Data Abstractions

Syllabus, Summer 2015

Information At-A-Glance

Instructor:	
Name:	**Adam Blank**
E-mail:	blank@cs.uw.edu
Office:	CSE 444
Office Hours:	Mon: 12:00pm – 1:30pm Wed: 12:00pm – 1:30pm Fri: 1:00pm – 2:00pm

Course Website:
http://cs.uw.edu/332
Visit early. Visit often.

Lecture
EEB 026 on MWF
10:50 AM – 11:50 AM

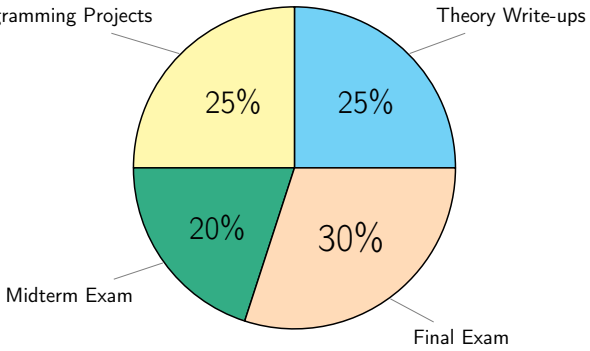
Textbook: Weiss, *Data Structures and Algorithm Analysis in Java*

Course Overview

This course covers fundamental data structures and algorithms. Prerequisite: CSE 311.
Covers abstract data types and structures including dictionaries, balanced trees, hash tables, priority queues, and graphs; sorting; asymptotic analysis; fundamental graph algorithms including graph search, shortest path, and minimum spanning trees; NP-completeness; concurrency and synchronization; and parallelism.

Assessments

Every assessment we give you has a very important purpose to your understanding of the material. Here's a handy pie chart that explains how your grade will be calculated:



Programming Projects. There will be *three* programming projects. The first one is worth half as much as each of the latter two. Programming projects will be graded on *correctness*, *architecture and design*, and *analysis*. Note that your answers to the analysis questions will be very heavily weighted. We will not grade you on code style, as long as your code is somewhat readable. Program design/architecture and analysis are crucial in this course, because we are studying data structures and the tradeoffs made during algorithm/data structure design.
Theory Write-ups. There will be *eight* "theory write-ups". These will each be weighted equally and will directly test your understanding of topics we are covering and the theory behind them. We require these to be turned in electronically. We urge you to learn LaTeX, but we will accept scanned written assignments as well.
Exams. We will have one midterm and one final exam. Both of these will be held during class. See the course website for more details.

Late Policy

You will have **three** "late days" which you may use for any assignment or project (except the ones due on the last day of class). Each late day allows you to turn in an assignment up to 24 hours late. Note that you may not use more than two late days on any particular assignment. If you are working on a partner's project, you may only use a late day if *both* group members have one remaining.

If you have used up your late days, you will incur a penalty of 10% of per 24 hours late. If unusual circumstances 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 youre sick in bed at home, you should still be able to make a phone call or send an email.) If you contact the instructor well in advance of the deadline, we may be able to show more flexibility in some cases.

Extra Credit

We will keep track of any extra features you implement (the Above and Beyond parts). You won't see these affecting your grades for individual projects, but they will be accumulated over all projects and used to bump up borderline grades at the end of the quarter. The bottom line is that these will only have a small effect on your overall grade (possibly none if you are not on a borderline) and you should be sure you have completed the non-extra credit portions of the homework in perfect form before attempting any extra credit. They are meant to be fun extensions to the assignments.

Getting Help

Please don't be afraid to ask for help if you don't understand something. Adam holds *at least three* office hours a week, and he gets lonely and bored if you don't show up! He also shows up early to lecture and is happy to answer any questions you might have before or after lecture.

At office hours, you can ask for clarification on a lecture (or for a *repetition* of the lecture!). You can ask for help with a frustrating part of the homework. You can even show up just to tell us you're frustrated and vent.

Here's some first steps on how to get help:

- Come to office hours
- Ask someone on course staff questions before/after lecture, before/after section, etc.
- Post on Piazza asking a question

Collaboration & Academic Integrity

Some programming assignments will be "partner assignments" in which you will work closely with another student. For all other assignments, we expect **all written/programmed work** to be your own. You may not look at someone else's write-up or code (even after it was due).

You must at least attempt a problem on your own before discussing it in a group–but we do encourage you brainstorm together! During brainstorming sessions, you may use a whiteboard, but you may not take any written or photographed work outside of the session. **If you collaborate with anyone in *any* capacity, you *must* identify them at the top of your assignment as a collaborator.**

If you do not follow these rules, you will be considered to have cheated. Cheating is a very serious offense. If you are caught cheating, you can expect a failing grade and initiation of a cheating case in the University system. Cheating is an insult to the instructor, to the department and major program, and most importantly, to you. If you feel that you are having a problem with the material, or don't have time to finish an assignment, or have any number of other reasons to cheat, then talk with the instructor. Copying others' work is not the solution.

To avoid creating situations where copying can arise, never e-mail or post your solution files. You can post general questions about interpretation and tools but limit your comments to these categories. If in doubt about what might constitute cheating, send the instructor email describing the situation. For more details see the Academic Misconduct web page.

Computing Resources

We will use Java (7 has been tested, 8 will also probably work fine) for programming assignments. JGrasp is a great editor for introductory programming–but you are no longer an introductory programmer! It's time to graduate to a real IDE. We don't particularly care which one you use (Eclipse, IntelliJ, NetBeans, Vim, Emacs, etc), but we will only *support* Eclipse (and vim). This means if you use a different editor and you have questions about why it's not working, we can't guarantee we will be able to help.