CSE 332: Data Abstractions

Course Syllabus, Summer 2017

Information At-A-Glance

Instructor:	
Name:	**Evan McCarty**
E-mail:	ejmcc@cs.uw.edu
Office:	CSE 214
Office Hours:	Mon: 11:00am – 12:00pm Wed: 11:00am – 12:00pm

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

Lecture
EEB 005 on MWF
09:40AM – 10:40 AM

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

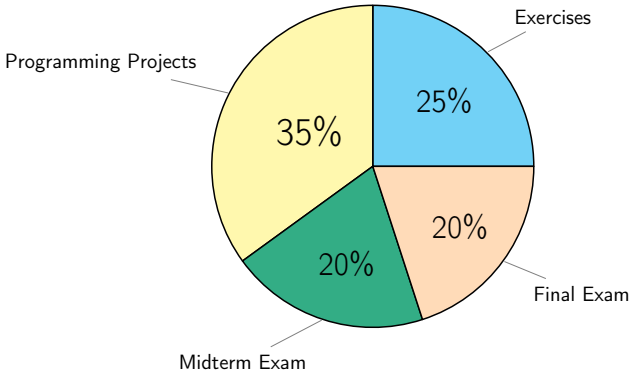
Course Overview

This course covers data structures, algorithms, and parallelism. 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; multithreading and parallel algorithms; P and NP complexity classes.

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. 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 and you follow the guidelines explicitly given in the project handouts. Program design/architecture and analysis are crucial in this course.

Exercises. There will be approximately weekly "exercises". 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. The midterm will be held in lecture.

Late Policy

You will have **four** "tokens" to use over the course of the quarter in one of two ways:

- 24 extra hours for a project (this is a standard late day)
- a re-submission of an exercise after getting the grade back

If you use a token on a partners project, all members must use a token, which means that if only one of you have one, you may not submit late. Projects will not be accepted late after tokens have run out. Leftover tokens will not contribute to your grade in any way.

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 you're sick in bed at home, you should still be able to 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 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 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 8 for programming assignments. We don't particularly care which IDE you use (Eclipse, IntelliJ, NetBeans, Vim, etc.), but we will only *support* Eclipse. 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.