CSE 332
Data Structures and Parallelism
Autumn 2016

Instructor:	Ruth Anderson
Email:	rea at cs.washington.edu
Office Location:	Allen Center (CSE) 460

Course Web Page: http://www.cs.washington.edu/332/

Description: 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. **Prerequisites:** CSE 311

Course Text: *Data Structures and Algorithm Analysis in Java* 3rd Ed., Mark Allen Weiss, Addison Wesley: 2012, ISBN-10: 0132576279. (We will also try to support the 2nd edition: Addison Wesley: 2007, ISBN: 0-321-37013-9.) Our course calendar will list sections of the textbook that are most relevant to the topic discussed in class that day. As lecture does not provide enough time to cover all material in detail, we strongly recommend that you read the textbook to clarify topics and find more examples as well as to examine Java implementations of the data structures and algorithms discussed during lecture. Copies will also be on reserve at the Engineering library. We also recommend a Java reference book of your choice, one is: Core Java(TM), Volume I--Fundamentals, Cay S. Horstmann and Gary Cornell, Prentice Hall.) We will use a set of free on-line notes for the material on parallelism and concurrency.

Computing Resources: We will use Java 8 for programming assignments. We strongly recommend although will not require that you use the Eclipse development environment. Links for downloading and installing Java and Eclipse can be found on our course web page.

Communications: The course Piazza message board should be your first stop for questions about course content and assignments. Before posting, first check that your question has not already been answered on the message board, and if not, ask it there. If it is not possible to ask your question on the message board without revealing details of your solution, please send email to cse332-staff at cs.washington.edu, which will go to the instructor and TAs. In general we prefer that you send questions to the cse332-staff list instead of to an individual staff member so that you will get a faster response time and the entire staff can remain aware of questions and issues. 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 (only course staff will post to the course email list).

Assignments: Assignments will be a mix of written/typed exercises and programming projects:

> **Exercises**: There will be approximately weekly "exercises". These will directly test your understanding of topics we are covering and the theory behind them. We require these to be turned in electronically. LaTex is one option, but neatly written scanned documents will also be fine as long as they are readable. In some cases programming will be required. You will submit exercises *individually*.

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 readable and you follow the guidelines explicitly given in the project handouts. Program design/architecture and analysis are crucial in this course. Except under unusual circumstances, projects will be done in *pairs of two students*.

Exams: We will have one midterm exam during lecture (date TBA) and a final exam at 8:30-10:20am Thursday December 15, 2016. 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: 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 programming project, all members must use a token, which means that if only one of you has 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.)

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

- 25% - "Written" Exercises
- 25% - Programming Projects
- 20% - Midterm Exam
- 30% - Final Exam

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.

Collaboration & Academic Integrity: Most programming projects will be "partner assignments" in which you will work closely with another student. For all other exercises, 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 to brainstorm together! During brainstorming sessions, you may use a whiteboard, but you may not take any typed/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.

Referring to solutions found on the web or solutions from this or other courses from previous quarters is also considered cheating. We plan on running similarity-detection software over all submitted student assignments, including assignments from past quarters.

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.](#)

CS 332 – First Day Assignments

0) **Review Java and explore Eclipse** – Instructions for installing on your home machine are on our course web page. Note: we will use Java 8. Now would be a good time to review material from CSE 143!

1) **Project #1** – Your first programming project will be posted **soon**. Look for an email announcing its arrival to verify you are on the course mailing list! Please fill out this Partner Request Survey (https://grinch.cs.washington.edu/partners) (preferably) by the end of Wednesday. Read more about working with partners [here](#). Two people may request to work together (both must request this on the survey) but we also have a research-based tool that will assign partners for you. We want you to be able to get started on the project ASAP as there will be some project deadlines early next week.

2) **Preliminary Information Survey**: Please fill out the preliminary survey posted on our course web page by the evening of **Thursday Sept 29th.**

3) **Reading** in *Data Structures and Algorithm Analysis in Java*, by Weiss (Reading is the same in both 2nd Edition and 3rd Edition)
- For this week:
    - › (Today) Weiss 3.1-3.7 –Lists, Stacks, & Queues (Topic for Project #1)
    - › (Fri) Weiss 2.1-2.4 –Algorithm Analysis
    - › (Also useful) Weiss 1.1-1.6 –Mathematics Review and Java