CSE 332

Data Structures and Parallelism

Autumn 2020

Instructor:	Ruth Anderson
Email:	rea at cs.washington.edu
Office Location:	Allen Center (CSE) 558 (not during 20au!)

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 (11, 12 or 13 will likely work) for programming assignments. We strongly recommend although will not require that you use the IntelliJ development environment. Links for downloading and installing Java and IntelliJ can be found on our course web page.

Communications: The course 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 either use a *private* post on the message board or 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 smaller exercises and larger 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. Some exercises require you to write short bits of code or complete an activity on line, while others are more traditional "written" exercises. For "written" exercises, 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. Exercises are to be done ***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***.

Quizzes: Due to the online-only nature of this offering, we will not be holding a traditional in-person midterm and final exam. Instead, we will have a total of 3 take-home, open-book quizzes spaced out during the quarter. You will be allowed to work together in groups on these quizzes. Each quiz will be released Wednesday afternoon on Gradescope and will be due Friday afternoon. More details including the exact release and due times will be announced later.

Late Policy: **Exercises and Quizzes may not be turned in late**. You will, however, have four late days that may be used to gain an extra 24 hours for a programming project. If you have used up your late days, a penalty of 10% per day will be assessed. **Programming projects will not be accepted beyond 48 hours from the original due date.**

Late days are intended to help you manage your time and life circumstances; leftover late days will not contribute to your grade in any way. If there are extenuating circumstances that prevent you from submitting an assignment on time, you should discuss this with the instructor (preferably in advance).

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

- 25% - Exercises
- 40% - Programming Projects
- 30% - Quizzes
- 5% - Participation

Participation: Your participation grade includes completing surveys and participation in activities during lecture and section, as well as other course activities. We realize that we are likely to have a number of students taking CSE 332 in 20au who will not be able to attend lectures or section for a variety of reasons (e.g. being in a different time zone, family or job responsibilities). While the course is designed for synchronous participation - we will have some in-class activities during lecture and section, students who cannot attend lecture will still be able to participate in these outside of lecture. Synchronous attendance at lecture and section is STRONGLY encouraged, but will not be "required".

Extra Credit: We will keep track of any extra features you implement for programming projects (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: Programming projects will be “partner assignments” in which you will work closely with another student. For 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 after first attempting on your own, 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.

Zoom and Privacy:

This course is scheduled to run synchronously at your scheduled class time via Zoom. These Zoom lecture sessions will be recorded. The recording will capture the presenter's audio, video and computer screen. Student audio and video will be recorded if they share their computer audio and video during the recorded session. The recordings will only be accessible to students enrolled in the course to review materials. These recordings will not be shared with or accessible to the public. Note: Quiz Sections and Office Hours will be held over Zoom but NOT be recorded.

The University and Zoom have FERPA-compliant agreements in place to protect the security and privacy of UW Zoom accounts. Students who do not wish to be recorded should:

- Change their Zoom screen name to hide any personal identifying information such as their name or UW Net ID, and
- Not share their computer audio or video during their Zoom sessions.

Disability Resources

The Disability Resources for Students (DRS) is a unit within the Division of Student Life and is dedicated to ensuring access and inclusion for all students with disabilities on the Seattle campus. They offer a wide range of services for students with disabilities that are individually designed and remove the need to reveal sensitive medical information to the course staff. If you have a medical need for extensions of assignment deadlines, these will only be granted through official documentation from DRS. Browse to this link to start the process as soon as possible to avoid delays.

You can refer to the university policies regarding Disability Accommodations for more information.

Religious Accommodations

Washington state law requires that UW develop a policy for accommodation of student absences or significant hardship due to reasons of faith or conscience, or for organized religious activities. The UW's policy, including more information about how to request an accommodation, is available at Religious Accommodations Policy. Accommodations must be requested within the **first two weeks of this course** using the Religious Accommodations Request form.

Extenuating Circumstances and Inclusiveness

We recognize that our students come from varied backgrounds and can have widely-varying circumstances. If you have any unforeseen or extenuating circumstance that arise during the course, please do not hesitate to contact the instructor in office hours, via email, or private Piazza post to discuss your situation. The sooner we are made aware, the more easily these situations can be resolved. Extenuating circumstances include work-school balance, familial responsibilities, military duties, unexpected travel, or anything else beyond your control that may negatively impact your performance in the class.

Additionally, if at any point you are made to feel uncomfortable, disrespected, or excluded by a staff member or fellow student, please report the incident so that we may address the issue and maintain a supportive and inclusive learning environment. Should you feel uncomfortable bringing up an issue with a staff member directly, you may consider sending anonymous feedback or contacting the Office of the Ombud.

CS 332 – First Day Assignments

0) **Fill out One or Two Surveys for 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! We want you to be able to get started on the project ASAP as there will be some project deadlines next Tuesday(!). We will be doing programming projects with partners. Read more about working with partners here: https://courses.cs.washington.edu/courses/cse332/20au/handouts/partners.pdf. You can be in a different lecture or a different quiz section from your partner. Only groups of size 2 will be allowed.

a) **Partner Request Survey** (https://grinch.cs.washington.edu/partners) - **EVERYONE** must fill this out the by **5pm TOMORROW, Thursday 10/01**. Two people may request to work together (both must request this on the Partners Request Survey). We also have a tool that will assign partners for you (see below). Note: **EVERYONE** must fill out the Partners Request Survey, whether they are requesting to work with a particular person or requesting to be added to the pool.

b) **Group Info Survey** (https://grinch.cs.washington.edu/groups) - If you want to be added to the partners pool and have us assign a partner for you, then you also must fill out the Group Info Survey by **5pm TOMORROW, Thursday 10/01** as well. This survey asks you to list the times that you are available to meet with your potential partner. If you will be in a different time zone in 20au, please be sure you list Seattle times that you are available in this survey.

1) **Review Java and explore IntelliJ** – Instructions for installing on your home machine are on our course web page under "Handouts" or can be found directly here: https://courses.cs.washington.edu/courses/cse332/20au/handouts/intellij.pdf. Note: we will use Java (11, 12 or 13 will likely work). Now would be a good time to review material from CSE 143!

2) **Exercise 1:** Your first exercise (https://grinch.cs.washington.edu/cse332/guessing-game) is a good way to review Java. **Due Monday 10/05 at 11:59pm**

3) **Pre-Course Check-In Survey, About-you Survey, and Introductions**: We would like to check-in with how you are doing during this unusual quarter and also get to know you a bit more. Please try to do all of these things by the evening of **Friday 10/02**

a) If you have not already done so, please fill out the Pre-Course Check-In Survey that was emailed before the first day of class.

b) Please also fill out this "About you" Survey so the course staff can learn more about you.

c) We would also like to be sure you know how to post questions on our course message board, and say "hi" to the rest of the class. Please post a short introduction on our course message board on the thread labeled with your quiz section.

4) **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