CSCI 232: Data Structures and Algorithms – Fall 2020

Course information

Meeting time/location:

Lecture: M/W 10:00-10:50PM http://wheelerlab.org/csci232

Labs: (one of)

Th 11:00-12:50 (section 1)

Th 3:30-5:20 (section 2)

Both sections remote - Microsoft Teams (Z-LAB--main-room)

Social Science 344 is available for Lab sections – please ask me

Course material/submissions/grades are in Moodle (http://umonline.umt.edu)

Instructor information

Instructor: Travis Wheeler
Office: Social Science 420
E-mail: travis.wheeler@umontana.edu
Phone: 406-243-6219

Office Hours: (http://wheelerlab.org/officehours)

Mon 2:00 – 3:30

Thu 2:00 – 3:30

Or by appointment (http://wheelerlab.org/schedule)

Teaching Assistants:

Trent Schweitzer
E-mail: trent.schweitzer@umontana.edu
Office Hours: Tue 11:00 – 1:00
Location: in Teams – "TA Office hours"

Nate Johnson
E-mail: nathan5.johnson@umconnect.umt.edu
Office Hours: Fri 9:00 – 10:50
Location: in Teams – "TA Office hours"

Course Objectives

The purpose of this course is to introduce you to essential data structures and algorithms that will serve as valuable building blocks for the remainder of your career as a computer scientist. In this class, we emphasize understanding of both (i) the methods for implementing fundamental data structures and algorithms and (ii) the ways in which these data structures algorithms can be used in code you will write for the remainder of your career. You will:

- Become familiar with fundamental data structures like stacks, queues, priority queues, associative arrays / hashes, and graphs (e.g. search trees and perhaps tries)

- Become familiar with fundamental algorithms based on these data structures, including sorting, clustering, graph search, and string search
- Improve your facility with software development, by implementing these data structures and algorithms in Java
- Become familiar with the basic notion of run time and space analysis, as applied to algorithm development

Other courses required

Prerequisite: CSCI 136 / 152
Corequisite: M 225 or M 307 (in prep for Analysis of Algorithms, CSCI 332)

Required textbook

Algorithms Fourth Edition
By Robert Sedgewick and Kevin Wayne
Booksite: http://algs4.cs.princeton.edu/

Flipped classroom

In this class, we turn the "talking-head" part of instruction (i.e. the lectures) into something you can do at home, and make our time in class into a more engaged experience. You will watch video lectures online before class, much as you might be expected in other courses to read a text book. Our class time is devoted to a combination of discussion, individual and group problem solving, and instructor-led clarification of some of the complex ideas in the material.

This approach places a clear burden on you:

- Before class, watch the recorded lectures and skim the assigned pages in the book. I will assume that you have done so. It will be apparent if you arrive in class without having watched the assigned lectures. Why? Because a decent portion of class time will be spent with activities such as:
    - Talking with classmates (and instructors) about the day's material
    - Working on problems in small groups
    - Answering questions from, and building on concepts presented in, the lectures
    - Also … there will be "quizzes" in moodle due before class
- After class
    - Review any of the lecture videos that remain confusing
    - Read the text in detail

You will be expected to watch a subset of the recorded lectures available within these Coursera courses:

Part 1: https://www.coursera.org/learn/algorithms-part1
Part 2: https://www.coursera.org/learn/algorithms-part2

Expected viewing for each lecture session will be found on moodle.

Lab

Most weeks, you will be expected to "attend" a ~2 hour lab section. Specific activities will vary from week to week, but in general you will be expected to implement and experiment with some basic data structure or approach that we have discussed in class. These lab sessions are intended to give you hands-on experience with the structures we care about, and will lay

the foundation necessary to succeed in written and programming assignments. You will be expected to submit the results of your in-lab work, usually with a brief writeup.

Schedule

Below is an ordered set of topics I expect to cover. It is subject to change. Please consult moodle for up-to-date schedule and reading assignments. Lectures will cover the reading material as comprehensively as possible. Students are expected to supplement lectures with a careful study of the relevant sections of the textbook.

- Fundamentals (Objects, data types, APIs, Analysis, Stacks, Queues)
- Sorting/Selection (Elementary, Mergesort, Quicksort, Priority Queues)
- Searching (Symbol Tables, Search trees, Hash tables)
- Graphs (Directed and undirected, BFS/DFS, Spanning trees, Shortest paths)
- Strings (Tries, Substring search)

Grading

Assignments will include both problem sets (questions requiring written answers) and programming assignments.

Problem sets:	25%
Programming:	25%
Lab work:	15%
Exams:	25%
Quizzes:	10%

The "curve": you may have heard that the grades assigned on my exams and assignments are often quite low. I account for this, and set grade cutoffs accordingly. Cutoffs are usually lower than the typical 90/80/70 splits. I will provide an update with approximate cutoffs as the semester progresses.

Exam Schedule (This will not be popular; it's also possible that it will change)

There will be two "mid-term exams" and one optional and cumulative final exam, all contributing equally to the total 25% exam component of the course grade. The exam will be oral, in the form of a one-to-one meeting with me: during

Midterm dates (approximate):

- during the week of Sept 28-Oct 2
- during the week of Nov 9-13

Final exam:

- during the week of Nov 19-25

Cheating

Academic dishonesty (including plagiarism) will not be tolerated. Cheating hurts all involved:

- It devalues the grades earned by others in the class, and the degree from our program
- It leaves you without the skills you've asked (and paid) me to help you gain
- (Actually, it doesn't hurt me; it just annoys me, because of the two above points)

Consult the university's student conduct code for more details. I will follow the guidelines given there. I will seek out the maximum allowable penalty for any academic dishonesty that occurs in this course. If you have questions about what constitutes acceptable use of resources, please feel free to reach out to me – I'll respect your attempts to understand the ethics of being aware of the work of others.

Specifically, do not search for answers online. I'm not naïve enough to think these don't exist, and I have caught >25 people plagiarizing from the web in the past five years. Nearly all have received a failing grade. Also, don't copy solutions from your classmates.

This is not an idle threat: a graduate student working with me has developed software for identifying online plagiarism, and we will apply that software to your submissions. Similarly, I am intimately familiar with the sorts of solutions to written questions available online, and typically recognize these. I retain the right to question you about the material turned in. If it is evident that you don't understand what you turned in, I will reduce your score, and may treat your submission as an instance of cheating.

What's better than using online resources or copying from a friend? Seeking help from the TA or me. **Use office hours. Really.**

A caveat. Throughout the course of this class, you are encouraged to work together in small groups. This is because the best way to understand the subtleties of the homework problems is to talk (argue?) about the answers. Read below for more thoughts on how these goals ("don't plagiarize", but "do talk to each other") interact in various contexts.

Working together (problem sets)

I expect that most of you will end up talking about class assignments with other students – that's good! Each of you should work on all of the problems independently, and not just sub-divide the questions among group member. You are welcome to discuss problems and collectively devise solutions at the conceptual, but you should not share the way you've written up your solution – each of you should independently write up a separate submission. Do not write your solutions up then share them with someone else. Though the ideas behind your solutions may be similar, the text should be your own – demonstrate your command of the problem with a personalized solution.

(In other words: don't be a leech and let your partner do all the work. Unless you learn how to solve problems, you will get burned on the exams and thus for your final grade. Even if you somehow get away with it, you won't learn the material, which will harm you in future classes and employment)

Working together (programming assignments)

I encourage discussion with others regarding programming assignments, as well. As with problem sets, these should be high-level discussions. Code should be written independently. If I suspect copying or plagiarism, I will ask you to explain each piece of the code to me, possibly resulting in a reduced grade or removal from class.

Working together (labs)

The lab sessions are where you have direct, hands-on exposure to implementing and using simple algorithms and the data structures they depend on. In lab, I want you to talk to each other, discussing the most minute details. This can (and should) involve looking at the code of your classmates. My goal with lab is that you learn the material and share what you've learned with others. In other words: I'm not worried about plagiarism here … except: when a lab asks that you write something about what you've done, that should be in your own words!

Late policy

Submissions for programming and homework assignments are due at the beginning of class. Late submissions will not be accepted. Every student will get one free extension on an assignment (programming or homework) for up to a week. You do not have to ask for this – just write that you are using your free extension when you turn it in. Don't waste this extension or feel obligated to use it; another extension will be given only in exceptional circumstances.

Attendance

Attendance is required. You are responsible for all material presented in class; some of that material is not covered in the textbook.

Computers

You may develop your programs on any machine that you like: we encourage you to use your own equipment. In the first lab, we will provide instructions for setting up a Java and terminal programming environment under Windows, Mac OS X, and Linux. Windows laptops will be available during lab sections.

Disabilities

Students with disabilities are encouraged to meet with me to discuss *any* accommodations they require.

Electronic devices (seems irrelevant for this remotely-taught class)

Turn off your cell phone during class. Students texting during class will be asked to leave. If you believe you may receive an urgent call during class, sit near the door, and be prepared to take the call outside the classroom.

Personal contact

I hope to establish as much personal contact with each of you as is possible in a class this size. Don't be afraid to visit my office hours, or ~~stop by my office~~ ping me on Teams to ask questions or say hello. In fact: I require that you do so at least once, in the first four weeks of class. Please visit during office hours, or schedule an appointment using the link on the first page of this syllabus. Expect to spend 15 minutes with me. Failure to do this will result in a 10% reduction in your grade (!)