CS 271: Data Structures
Spring 2022

Instructor:	Michael P. Rogers
Office:	Halsey 214
Office Hours:	MW 3:45-4:45 PM, TuThF 3:00-4:00 PM, Other Hours By Appointment
Email:	rogersm@uwosh.edu
Phone:	920-424-1388 / 309-825-6454
Discord:	https://discord.gg/GY2PHz2Q
Class Times:	MoWeThFri 11:30 AM-12:30 PM (MoWeTh: Halsey 309; Fr: Halsey 101C)
Credits:	4

Prerequisites: CS 262 with a grade of C or better.

Description: A course surveying the fundamental methods of representing data and the algorithms that implement and use those data representation techniques. Data structures and algorithms include; linked lists, stacks, queues, trees, heaps, priority queues, hashing, searching, sorting, data compression, graphs, recursion. Analysis topics include: elementary big-O analysis, empirical measurements of performance, time/space trade-offs, and identifying differences among best, average, and worst case behaviors.

Course Website: if it happens in this course, it will be posted on UWO's Canvas site. Set up notifications to be alerted when announcements, grades, assignments, notes, etc., are posted.

Required Textbook:

- Goodrich, Tamassia and Goldwasser. *Data Structures & Algorithms in Java,* 6th Edition. Wiley, 2014.

Course Outcomes :

Upon successful completion of the course, students will be able to:

1. Given a non-recursive algorithm, the student will be able to examine its loop structures and infer its asymptotic runtime using big-O notation.
2. Given a recursive algorithm, the student will be able to examine its recursive structure, determine the corresponding recurrence relation (from a small collection of commonly occurring recurrence relations), and use the recurrence relation in determining the asymptotic runtime of the algorithm using big-O notation.
3. Given the description of a computational problem requiring a mixture of search, insertion, and/or deletion operations on collections of data, the student will be able to compare the relative advantages of using arrays and linked lists in solving the problem efficiently.
4. Given a classical computational problem (e.g., infix-to-postfix conversion, postfix-expression evaluation, path planning, minimum-spanning tree computation), the student will be able to trace a solution to the problem using appropriate data structures (e.g., stacks, queues, binary trees, binary search trees, red-black trees, graphs) and to predict the asymptotic runtime of the solution based on the selected data structures.
5. Given a collection of unordered data, the student will be able to trace the execution of an advanced sorting algorithm (such as quick sort and heap sort) on this data set.
6. Given a set of data keys, the student will be able to trace through a sequence of key insertions, searches and deletions on a balanced tree structure. The student will also be able to discuss the relationship between the number of keys and the execution time of these operations.

7. Given a set of data keys, a hash function, a table size, and a collision-handling strategy, the student will be able to trace through a sequence of key insertions and searches, and to discuss how varying the table size, hash function or collision-handling would affect the execution time of these operations.
8. Given a graph data structure, the student will be able to implement it using either adjacency lists or an adjacency matrix, to traverse it using either a depth-first or breadth-first strategy, to identify its structural properties (whether it is directed, cyclic, connected, complete), and to trace the execution of one or more classical graph algorithms (e.g., Dijkstra's shortest path, topological sort or minimum- spanning tree computation).
9. Given a problem requiring the efficient use of a variety of data structures, the student will be able to apply object-oriented design principles in implementing and testing a solution to that problem in an appropriate object-oriented language.

Grading Criteria:

Category	%
Exams (3)	45
Labs (~12)	15
Assignments (~3-4)	25
Mini quizzes (~10)	10
Reading quizzes	5

Grade Scale:

%	≥ 92	90-92	88-90	82-88	80-82	78-80	72-78	70-72	68-70	62-68	60-62	< 60
Letter	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

Exams: There will be 3 equally weighted exams, administered on or about

- Wednesday, March 2
- Friday, April 10
- Friday, May 13

You will be permitted to bring in one 8.5x11 inch sheet of hand-written, double-sided notes for each, but no other resources.

Labs: Labs will be assigned each Friday, and due the next Thursday by **11 PM**. Projects must compile in order to receive a non-zero grade. Do not wait until the last minute to attempt these, as what may seem easy in the abstract may in fact not prove to be the case. Late lab submissions will not be accepted, so do not wait until the last moment to hit the submit button!

Mini Quizzes: These short quizzes (~5-15 min) will be administered weekly.

Reading Quizzes: To ensure that you are actually reading and studying the text (DSAJ) short quizzes (~1-2 minutes), will be administered at the beginning of almost every class. Typically, the reading will be over material to be discussed in class that day. Readings will not be explicitly assigned, but rather implicitly specified in each day's quiz name, e.g., RQ 4.2.1 means that you will be quizzed over section 4.2.1.

Coding Standards: All code must adhere to the coding standards as described on the website (under Helpful Resources): think of it as preparation for the workforce, where this is de rigueur. Points will be deducted for violations.

Absences: It has been scientifically proven that the most significant factor for predicting student success is attendance (although whether this is truly causation or merely correlation is another question). Students should attend each and every class as scheduled, and notify the instructor ahead of time if you will be absent, otherwise you risk missing a) some really great, Trevor Noah-level jokes, and b) reading quiz points. If you are unable to take an exam/mini quiz at the prescribed time a) let me know in advance, via email, and b) provide justification (a note from medical professional who treated you, or a representative from the Dean of Students Office).

Academic Integrity: The purpose of this course is to teach you in particular about data structures, and more generally how to think, how to *problem solve*. For that reason, the work that you turn in must be your own. You may have *general* conversations with students to clarify the nature of an assignment, and you can ask for help with debugging, but that second-set-of-eyes-student should not be looking at their code while they assist you. Sometimes students are unaware of whether or not they have committed plagiarism, so here are some tips:

1. if your problem solving begins with ctrl-C and ends with ctrl-V, you have committed plagiarism.
2. if your problem solving starts at Google and ends at Chegg, you have committed plagiarism.
3. if your problem-solving involves surreptitiously glancing at the exam of the student next to you and doing a virtual copy-and-paste, you have committed plagiarism.
4. if that tiny voice inside your head, the same one that inconveniently shuts up entirely while you are working on a data structures assignment, starts making "ahem" noises, you have committed plagiarism.

Let us consider the pros and cons of committing plagiarism.

Pros:
1. You have completed the assignment.

Cons:
1. You will have missed that exhilarating, ego-boosting, delicious eureka! moment that everyone experiences when they have, on their own, solved a difficult puzzle.
2. You will be caught, receive a 0 on the assignment/exam, and may face disciplinary action in front of a bevy of grim-faced administrators who you do *not* want to meet.

So what do you do when you can't solve a problem? The answer is simple, ask your instructor (or lab assistant) for help. We will use the time-tested Socratic method, asking questions that will lead you to the correct answer. Failing that, just write "I have no idea" in bold letters in the comment block at the top of the assignment, turn it in, and you will gain *some* points (more than 0).

For more detailed information on what constitutes academic misconduct, please see the discussion of UWS Chapter 14, Student Academic Disciplinary Procedures.

Accessibility: Your instructor is committed to ensuring a fair playing field. If you have a disability and need assistance (e.g., a note taker, certain seating, extra time to take tests, adaptive technology, etc.), please register with the Accessibility Center, and we work hard to accommodate your needs.

HATE HAS NO HOME HERE.

Non-discrimination and Anti-harassment: Your instructor is committed to maintaining a harassment-free, welcoming classroom, and will not tolerate discrimination on the basis of race, religion, creed, color, gender, identity/expression, ancestry, national origin, age, marital status, relationship to other employees, sexual orientation, disability, veteran's status, membership in the military, arrest/conviction record, political affiliation, or any other protected status.

Class Participation and Feedback: Your instructor relishes class participation and feedback. If you are lost, please, please ask, during class. You do not need to worry about slowing down the class, and your fellow students, who probably were thinking the same thing but were afraid to ask, will silently or possibly aloud applaud your efforts, as will I.

Rules for Success:
Apart from the usual -- work hard, study, ask early when you need help, don't cheat, get a good night's sleep -- here are three that need emphasis:

1. Read the book, **carefully**. It is a *masterpiece* of clear exposition.
2. When solving a problem, don't even *think* about writing code until you have drawn a picture visualizing the situation and a potential solution.
3. Come to class engaged, curious, ready to learn.

Masks: All students are required to wear an appropriate mask that covers their mouth and nose when they are in the classroom. They must also adhere to additional expectations communicated by the instructor or posted in the classroom. Note: UWO procedure dictates that, during the COVID-19 pandemic, an instructor cannot begin class until all students are wearing a mask properly. If a student is non-compliant with the masking policy and refuses to leave the classroom promptly when requested, the instructor is required to cancel class. Students responsible for class cancellation for these reasons will be referred to the Dean of Students office, and the student will be unable to attend class until they meet with the Dean of Students. The student may be dropped from the class by the Dean of Students.

Consumer Information: Students are advised to see the following URL for disclosures about essential consumer protection items required by the Students Right to Know Act of 1990: https://uwosh.edu/financialaid/consumer-information/

Note: In the event of any substantive changes to this syllabus, you will be notified in a timely manner.