CS2240 OL / Data Structures and Algorithms / 2023 Fall

Compiled: 2023-09-25 14:38

Instructor: Clayton Cafiero
Email: cbcafier@uvm.edu
Office: Innovation E309
Office hours: T 08:30–09:45 AM, Th 04:30–05:45 PM, F 09:30–11:00 AM, or by appointment.

GTA: Omar Awajan
Email: oawajan@uvm.edu
Office hours: T 02:00-04:00 PM
Location: Innovation E344 or via MS Teams

UTA help desk: S–Th, 5:00–7:00 PM
Locations: S/M Votey 207; T/W/Th Innovation E210

Prerequisites: CS 110 with a grade of C- or better

Overview: Design and implementation of linear structures, trees and graphs; examples of common algorithmic paradigms; theoretical and empirical complexity analysis; sorting, searching, and basic graph algorithms; common ways of storing data and the use cases for each, including stacks, queues, trees, heaps, and hash tables. Knowledge acquired will be applied in a multi-part programming project, and demonstrated in weekly quizzes and final exam.

Learning objectives:

- You will gain knowledge of fundamental data structures and algorithms, and recognize the use cases for each.
- You will gain understanding of the correctness of algorithms and their complexity.
- You will demonstrate knowledge of essential computer science concepts and skills through project programming (AOE CSES 3.3.1).
- You will understand how storing data in hardware affects the complexity of algorithms and determines how best to store the data in memory (AOE CSES 3.4.2).
- You will perform algorithm analysis using asymptotic notation to evaluate best-, average-, and worst-case space and time complexity (AOE CSES 3.7.5).

- You will collect, aggregate, clean, and model data in your project, in which you will analyze and compare different data structures and algorithms (AOE CSES 3.6.1).
- You will demonstrate basic proficiency in C++, including object-oriented design, abstraction, and recursion, by going through the software development cycle in lectures and project programming (AOE CSES 3.7.1-3.7.4, 3.7.6).

Course materials: Textbook: *Essential Algorithms: A Practical Approach to Computer Algorithms* by Rod Stephens, Wiley, 2013. ISBN-13: 978-1-118-61210-1. Out of print, but ebook is available through Howe Library. Other materials supplied on Brightspace.

Software: You must have a **C++ compiler**. If you are using macOS or Linux you probably already have a C++ compiler installed. Procedures for installation of a C++ compiler vary substantially substantially by OS.

You should use an **IDE suited to C++ development**. CLion is recommended. CLion is available with a free student license.[1] You are welcome to use a different IDE, but video coding sessions done with CLion, and instructions for projects, *etc.*, are with respect to CLion. If you do not want to use an IDE, a reasonably current version of gcc/g++ should work fine in your command line interface.

Starter / scaffolding code will be released on GitHub, and you will make your submissions via GitHub. GitHub is a service for hosting Git source code repositories. Git is the most widely used source code management tool in the world. Accordingly you will need a **reasonably current version of Git** installed on your machine. If you are using macOS or Linux you probably already have Git installed. For information on installing Git, see: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.

Project: Over the course of the semester you will complete your project in five parts. Each part will involve implementation of various data structures and algorithms, testing, and preparation of a written report. *Work must be your own*. Any code or analysis not authored by yourself or an instructor *must be cited* in the project submission. Project parts (5) are weighted equally. There may be brief, preliminary assignments to serve as checkpoints or scaffolding for your work. Points earned for these submissions will be included as part of your overall project grade.

Part	Topic	
1	Selecting your data set; creating your custom class; loading data from file; creating an array of objects; performing a calculation on the array	W 2023-09-13 11:59 PM
2	Create queue and stack classes; load your objects into queue and stack; push and pop; analysis of results	W 2023-09-27 11:59 PM
3	Create BST, AVL and splay trees; perform insert, search and delete operations on each; analysis of results	W 2023-10-18 11:59 PM
4	Comparison of sorting algorithms (*e.g.*, bubble, selection, merge, and heap); analysis of results	W 2023-11-08 11:59 PM
5	Creating hash tables, rehashing, comparing collision strategies; analysis of results	Th 2023-11-30 11:59 PM

Discussion: *You are encouraged to share your knowledge, discoveries, and ideas with your classmates, and to ask questions whenever you have them.*

[1] https://www.jetbrains.com/community/education

Final exam: The final exam will be open note / open book, but zero collaboration. The exam will be cumulative and will be administered on Brightspace during finals week 2023-12-11 through 2023-12-15. Precise format and scope TBD.

Quizzes: There will be 14 weekly, open book, open note quizzes, administered on Brightspace. Each quiz will concentrate on the most recent course material. Quizzes will have true/false, multiple choice, fill in the blanks, matching, ordering, and short answer questions. Quizzes will be available starting 12:01 AM each Monday, and will remain open until 11:59 PM the following Sunday. Quizzes will be timed—30 minutes—and you must complete the quiz in one sitting. (Quiz will be auto-submitted at 35 minutes.) You will have two attempts at each quiz, and your highest score will prevail. Be aware that some questions are selected from a pool of questions at random, so if you choose to make two attempts, you may not be asked all the same questions, and the order of questions may change. Questions within a quiz are weighted equally. Quizzes are weighted equally. I will drop your lowest quiz grade. You are not permitted to collaborate on quizzes and all answers should be your own.

Assessment: The course programming project is in five parts. This will assess:

- Your ability to use C++ templates, classes, and functions to create and analyze the data structures and algorithms taught in the course.
- Your ability to understand how various data structures work and to write an insightful analysis supported by the data you collect.
- Your ability to compare and contrast data structures and algorithms and to identify and justify the best implementation for a given problem.

Quizzes and the final exam will assess:

- Your knowledge of the concepts related to the data structures and algorithms, their properties, and how they function.
- Your ability to identify the time and space complexity and other properties of algorithms.
- Your understanding of algorithmic complexity as expressed in asymptotic notation.

Grading:

55%	project (5 × 110 points each)
26%	quizzes (14 × 20 points each, drop one)
15%	final exam
4%	survey participation (4 × 10 points each)
100%	TOTAL

Questions about grading: In order to receive consideration, if you have any questions or concerns about grades you must contact our graduate TA via email within one week of the grade release for any given assessment.

Early/late policy: If you submit a project component two or more days early, you will get 5 points extra credit. If you submit a project component one day early, you will get 2.5 points extra credit. Late submissions will be penalized 15 points per day, for up to three days. Late submissions will not be accepted after this period has elapsed. *Be careful not to omit files!* If a submission is found to have omitted files (code or report) you will be penalized 10% and contacted via email requesting that you supply the missing file(s). You will have 24 hours to submit any missing file(s). If you do not submit missing files within that time you will receive a 0 for the project component. Hint: Check your email regularly!

Course modules by week:

Week	Topics covered
Week 1	Welcome; modular arithmetic; recursive functions; intro to C++; output manipulation; file input
Week 2	Pointers, addresses, references; C++ templates; overloaded operators; node class for singly-linked lists
Week 3	Memory: heap and stack; node and stack classes; queues; intro to complexity; asymptotic notation
Week 4	More on complexity and asymptotic notation; loops; trees; traversal and search; binary search trees
Week 5	More on trees; AVL and splay trees; B-trees
Week 6	Binary heap; priority queues; intro to sorting; bubble sort
Week 7	More on sorting: selection, insertion, and merge sort
Week 8	Quicksort, bucket sort, radix, and heap sort
Week 9	Searching; Introduction to hashing; Horner hash; hash collisions; separate chaining
Week 10	Linear probing; rehashing; quadratic probing; double hashing
Week 11	Relations and equivalence relations; dynamic equivalence problem
Week 12	Disjoint sets: weighted union and path compression; introduction to graphs; topological sort; shortest path
Week 13	Dijkstra's algorithm; Bellman-Ford; network flows; max flow - min cut
Week 14	Max flow - min cut continued; Ford-Fulkerson and Edmonds-Karp; minimum spanning trees: Prim and Kruskal
Week 15	Review and final exam

Each module has its own section on Brightspace.

Academic integrity: The Department of Computer Science enforces UVM's Code of Academic Integrity. Any suspected violation of this policy will be referred immediately to UVM's Center for Student Conduct (https://www.uvm.edu/sconduct). Sanctions for a violation may include a grade of XF in the course. Additional violations can result in dismissal from the university. In a word: *Don't*. All students should be read and understand this policy. See: https://go.uvm.edu/cai.

Collaboration on quizzes and exams is strictly prohibited. Use of online services as a source of solutions is strictly prohibited. Using AI-content generators such as ChatGPT or websites such as Chegg or Course Hero to complete coursework is a form of academic dishonesty. Work you submit for an individual grade must be your own. Any work not produced by you (or teammates in the case of active learning exercises or labs, where applicable) must be cited. If you have any questions ask!

Any attempt to tamper with any autograder is a form of academic dishonesty. This applies wherever autograders are in use, for example on Brightspace or Gradescope.

Exams, quizzes, homework assignments, answer keys and solutions, presentations or lecture notes, specifications and rubrics are copyright protected works, unless clearly and explicitly indicated otherwise. Any unauthorized copying or distribution of protected works is a violation of federal law and may result in disciplinary action. This includes submission of protected works as prompts to generative AI. Sharing of course materials without the specific, express approval of the instructor may be a violation of the University's Code of Academic Integrity and an act of academic dishonesty, which could result in disciplinary action. Violations will be handled under UVM's Intellectual Property Policy and Code of Academic Integrity, as appropriate. See: https://go.uvm.edu/ipp and https://go.uvm.edu/cai.

Student course evaluations: Students are warmly encouraged to complete an evaluation of the course at its conclusion. Evaluations are anonymous and confidential, and the information gained, including constructive criticisms, will be used to improve the course.

Participation: Even though this is an online course, you are expected to be an active participant. The more engaged you are, the more you will learn—and the more fun you'll have. This includes reading assigned materials, watching instructional videos, and the like. Since we won't have regular face-to-face interactions in class, it's all the more important to ask questions—either in office hours or on an *ad hoc* basis. When it comes to asking questions, *please don't be shy!* There's no such thing as a "dumb" question (I earnestly believe this). If there's something you don't understand—*ask*! Asking questions helps you understand the material presented in the course. Also, when you ask a question you help me do a better job of explaining. If I explain something, and you still don't quite grasp it, chances are that I didn't do as good a job of explaining as I might have.

Student course evaluations: Students are warmly encouraged to complete an evaluation of the course at its conclusion. Evaluations are anonymous and confidential, and the information gained, including constructive criticisms, will be used to improve the course.

Defects / bonus points: As you might expect, we will deduct points on assignments, quizzes, or exams where you've made an error. It's only fair that instructors should be held to a similar standard. Accordingly, bonus points are awarded for reporting and correcting defects in instructor-written course materials.

- 1.0 point: Material defect. This includes any error whatsoever in code or solutions, or any error in writing that changes materially the sense of what is written. This also includes incorrect due dates for posted assessments.
- Up to 0.5 point: Minor defect. This includes typos, misspellings, or minor errors which do not affect materially the readability or sense of what is written. Determination of points for minor defects is at the instructor's sole discretion.

Due to the unfortunate behavior of some students who have favored the practice of "bonus point mining" over proper study, no student may earn more than three bonus points in this course.

Due to the fact that we are still writing and revising autograders, autograder defects are not fair game for defect bonus points (but by all means, please report, and perhaps we will award a discretionary point here and there).

Bonus points for any given defect are only awarded to the first student (across all sections) who correctly identifies the error *and provides a valid correction*. Bonus points are not available for any materials which are clearly marked as drafts. Bonus points are added to your final grade before assigning letter grades—a point or two may make a big difference. Happy hunting.

The secret word is "turnip."

Diversity, equity, and inclusion: UVM is a place where you should be treated with respect and kindness. We welcome individuals of all ages, backgrounds, beliefs, interests, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability, and other visible and non-visible differences. All students are expected to contribute to a respectful, welcoming and inclusive environment for every other member of the community. If you ever feel that you have been unfairly treated or judged by an instructor, a mentor, another student, or another member of the community, please let someone know. Your instructors and advisors in the CEMS Office of Student Services are available to discuss any concerns, or you can report an incident of bias through the bias report program (https://www.uvm.edu/deanofstudents/bias_response_program).

Conduct: Be kind to one another and to yourself. Be respectful of yourself, others, and the institution. Please arrive on time. Please, no food in class. Please, no cell phones in class (except for using the iClicker app when requested). You may use a laptop or tablet, but only for active learning sessions, pair programming, taking notes, or assistive technologies.

For other policies on classroom conduct, please see: https://go.uvm.edu/srr and https://go.uvm.edu/csc.

Accommodations: In keeping with UVM policy, if you have a documented disability and are interested in utilizing ADA accommodations, you should contact Student Accessibility Services (SAS), the office of Disability Services on campus for students. SAS works with students and faculty in an interactive process to explore reasonable and appropriate accommodations, which are communicated to faculty in an accommodation letter.

Contact SAS: A170 Living/Learning Center; +1 802 656 7753; access@uvm.edu; or visit https://www.uvm.edu/access.

If you are entitled to use the Exam Proctoring Center, please book reservations at least four days in advance.

Promoting health and safety: If you are concerned about a UVM community member or are concerned about a specific event, we encourage you to contact the Dean of Students Office (+1 802 656 3380). If you would like to remain anonymous, you can report your concerns online by visiting the Dean of Students website at https://www.uvm.edu/deanofstudents.

Wellbeing resources:

- Center for Health and Wellbeing: https://www.uvm.edu/health
- Counseling and Psychiatry Services (CAPS): +1 802 656 3340
- Food Insecurity Assistance: https://www.uvm.edu/health/food-insecurity-uvm

Student advocacy: https://www.uvm.edu/deanofstudents/student_advocacy

Your identity at UVM: Students at UVM can specify the first name and pronoun they want used on campus. For information on how to update your preferred name and personal pronouns as well as

keeping your legal name private, see: https://www.uvm.edu/registrar/name-and-pronouns. For UVM policy on lived name and pronouns, see: https://go.uvm.edu/lnpr.

Religious holidays: Students have the right to practice the religion of their choice. In order to receive extensions or excused absences, you should submit via email your documented religious holiday schedule for the semester within the first two weeks of class. Reasonable extensions will be granted where assignment deadlines conflict with religious holidays.

Student athletes: In order to receive extensions or excused absences, you should submit via email appropriate documentation as soon as possible, preferably within the first two weeks of class. Reasonable extensions will be granted where assignment deadlines conflict with team events or team travel.

Statement on alcohol and other drugs: We want you to get the most you can out of this course. Therefore, you are expected to familiarize yourself and abide by the University's policies with regard to alcohol, cannabis, tobacco, and other drug use. See: https://www.uvm.edu/sites/default/files/UVM-Policies/policies/drugandalco.pdf Please do everything you can to optimize your learning and to participate fully in this course.

Class format changes: The University of Vermont reserves the right to make changes in the course offerings, mode of delivery, degree requirements, charges, regulations, and procedures contained herein as educational, financial, and health, safety, and welfare considerations require, or as necessary to be compliant with governmental, accreditation, or public health directives.

Changes to this document: This document is subject to change. Any such change will communicated via class email and with an announcement on Brightspace. The latest version of the syllabus will always be available on Brightspace.