CS124OL / Data Structures and Algorithms / 2022 Fall

Compiled: 2022-08-28 10:53

Instructor: Clayton Cafiero <clayton.cafiero@uvm.edu>
Office: Innovation E309
Instructor office hours: W 09:00–10:00 AM; Th 12:00–01:00 PM; F 09:00–11:00 AM
- via Microsoft Teams / CS124OL1 Data Structures & Algorithms
UTA: Jason Lobell <jason.lobell@uvm.edu>
UTA office hours: T 12:00–01:00, via Microsoft Teams
Help desk: Su-Th 05:00–07:00 PM. Details TBA

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: 978-1118612101. Other materials supplied on Blackboard.

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] 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	Thursday, Sep 15 11:59 PM
2	Create queue and stack classes; load your objects into queue and stack; push and pop; analysis of results	Thursday, Oct 06 11:59 PM
3	Create BST, AVL and splay trees; perform insert, search and delete operations on each; analysis of results	Thursday, Oct 27 11:59 PM
4	Comparison of sorting algorithms (*e.g.*, bubble, selection, merge, and heap); analysis of results	Thursday, Nov 17 11:59 PM
5	Creating hash tables, rehashing, comparing collision strategies; analysis of results	Thursday, Dec 08 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.* Let's conduct course discussions on Microsoft Teams.

Final exam: The final exam will be open note / open book, but zero collaboration. The exam will be cumulative and will be administered on Blackboard during the last week of the semester. Precise format and scope TBD.

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

Quizzes: There will be 14 weekly, open book, open note quizzes, administered on Blackboard. Each quiz will concentrate on the most recent course material. Quizzes will have true/false, multiple choice, and short answer questions. There will be around 20 questions, though the number of questions may vary. Quizzes will be available starting 6:00 AM each Thursday, and will remain open until 11:00 PM the following Sunday. Quizzes will be timed—30 minutes—and you must complete the quiz in one sitting. 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.

Surveys: Surveys will be released on Blackboard throughout the semester, and I expect you to complete and submit them. Your responses are anonymous and I see only aggregate results. I can see whether or not you have completed the survey. No credit will be given for surveys after their deadline and there is no partial credit given for a survey. Surveys are weighted equally.

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 parts)
25%	quizzes
15%	final exam
5%	survey participation
100%	TOTAL

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

Early/late policy: If you submit a project component two or more days early, you will get 5% extra credit. If you submit a project component one day early, you will get 2.5% extra credit. Late submissions will be penalized 10% 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	Dijkstras 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 Blackboard.

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.

Defects / bonus points: As you might expect, I will deduct points on assignments, quizzes, or exams where you've made an error. I think it only fair that I should be held to a similar standard. Therefore, bonus points will be awarded to the student who correctly identifies any error for which I am responsible in materials I distribute to this class. This includes sample code, presentation slides, lecture notes, quizzes, *etc*. For any given error, a bonus point will be awarded only to the *first* student reporting the error. Bonus points are added to your final grade before assigning letter grades—a point or two may make a big difference. Happy hunting.

Other opportunities for bonus points may be announced in class.

Academic integrity: Materials used in this course, including, but not limited to assignments, specifications, rubrics, exams, quizzes, and instructional materials, are copyright protected works. Any unauthorized copying or distribution of course materials is a violation of federal law and may result in disciplinary action. 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. This includes, among other things, distributing course materials for the purpose of sharing or seeking solutions to homework or programming assignments. Use of online services for help or solutions is strictly prohibited. Any outside sources, where permitted with prior instructor approval, must be cited.

Submissions may be screened and evaluating using plagiarism detection software (MOSS).

By enrolling in this course, you acknowledge that you have read and understand the Code of Academic Integrity, and that you agree to abide by this code. Any suspected violations will be dealt with promptly. In a word: *Don't*.

See: https://www.uvm.edu/policies/student/studentcode.pdf for more information.

Statement on alcohol and other drugs: I 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.[2] Please do everything you can to optimize your learning and to participate fully in this course.

Accommodations: In keeping with University 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.

[2] See: https://www.uvm.edu/sites/default/files/UVM-Policies/policies/drugandalco.pdf

CEMS Inclusion Statement: I wholeheartedly support the CEMS policy on diversity, equity, and inclusion:

> "Our intention is for CEMS to be a place where you will 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 nonvisible differences. All members of the College 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 CEMS 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 Campus Bias Response Program."

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.