Syllabus
CMSC 262: Data structures and algorithms in application

Fall 2025

Time:	MW 4pm
Room:	Rotunda 356
Website:	`https://cs.longwood.edu/courses/cmsc262`
	`https://longwood.instructure.com/courses/1318363`

Introduces many of the classic advanced data structures and algorithms in the context of a survey of important applied fields of computer science. Topics include artificial intelligence, relational databases, and human-computer interaction. Prerequisite: CMSC 162; MATH 175 recommended. 3 credits.

Professor:	Don Blaheta
Office:	Rotunda 334
Phone:	x2191
Email:	`blahetadp@longwood.edu`
100% Office hours:	Mondays 3–4pm; Tuesdays 11am–noon; Thursdays 1–2pm; Fridays 3–4pm

Overview

Back at the Dawn of Time—around the mid-20th century—the field of computer science was all about number crunching. Increasingly in the last several decades, we've seen the research problems of computer science become much more about how to store, process, and interact with other kinds of information. Though not every computer scientist need be an expert in areas like artificial intelligence, databases, and human-computer interaction, one should have a general grounding in all of them.

Such is the aim of this class. We will cover a variety of disparate topics in computer science that will provide you with the tools you need, whether to continue in those subfields or just to understand what your colleagues are talking about.

Student Learning Outcomes

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

1. implement classic data structures and algorithms from pseudocode and descriptions;
2. reformulate problems in terms of search through an abstract problem space;
3. combine information from different sources using probabilistic models;
4. explain and address challenges faced in presenting information to diverse audiences; and
5. construct a simple relational database and write queries for it using standard notation.

Resources

There is no required textbook for this course.

This course surveys many different areas, and at the level of detail we need for now, there are many textbooks as well as free online resources that cover them sufficiently. Textbooks are expensive, and this is a good opportunity for you to be resourceful. I've built a website that maps appropriate readings in various books—in multiple editions—to the topics of the day, as well as links to Wikipedia and other websites with relevant reading.

You'll be given an account on the department Linux machines (if you don't already have one), and you'll do your programming work there. You will be expected to have a computer that can connect to the internet and various websites, and run PuTTY or another ssh client to connect to the department Linux machines.

You will need to join the CMSC Slack server and the channel for this course (`#cmsc-262`), and check it somewhat regularly.

In the hopefully unlikely event that you need to go into quarantine or isolation (for Covid-19 or for some other reason), but are otherwise well enough to continue working, I'll expect that you have a device (your computer, or a phone or tablet) that is capable of connecting to a live meeting via Zoom, and reasonable bandwidth to accommodate that.

AI Policy

My general feeling about AI is this: AI is a tool. Use it when it's helpful, don't use it when you could do it better or faster yourself.

That said, there are certain skills that programmers and computer scientists will need to develop and execute without the help of AI, slightly because AI might not *always* be available but mostly because you'll need to be able to evaluate and debug the code that the AI (or other programmers) have given you. Thus for assignments that are about *developing* your programming skills (labs, homeworks, projects), I'm going to discourage use of AI until you've given a few solid attempts without. For assignments that are *assessing* your skills (exams) I'll have specific instructions on whether you are or are not allowed to use generative AI to assist. *In general* tasks that you're doing on your own time will permit use of AI, but please attend to specific instructions on each assignment.

I will expect that when you *do* use generative AI, you will document it: say which AI system you used and what help it gave you. **In a comment or embedded link, you should include the "share" URL that lets others view your prompts along with the AI's responses.** Some assignments will have additional instructions how to document this.

(Note that although Longwood's Honor Code does not inherently ban the use of AI, some other professors seem to think it does, so for your safety you should check with each professor before using it in their class.)

Attendance, absence, and zoom (fka "Covid-19 notes")

I have a few policies that originally evolved in response to the pandemic but I've decided they're just good policy so I kept them. Here's the gist: It's really easy to keep zoom open for every class, and it's not nearly as good as in-person attendance but way better than total absence. So I open Zoom every day and ask you to make good choices.

Attending class. There are two ways you can attend class: in person, or via Zoom. Either mode of attendance is sufficient to be "attending" (but not necessarily engaging or participating). If you attend via Zoom,

- you must have a reason, and
- you must say what it is,

but I don't need any medical detail and if it's not directly illness-related I'm not going to police that. (Again: be an adult and make good choices.) The Zoom experience is nowhere near equivalent to the in-person experience and is not a replacement for it; but if you are quarantined, or otherwise just can't attend in person on a particular day, zooming is better than total absence.

Zooming vs masking. Although we've moved to "endemic" on Covid-19, remember that masking is still a tool in our toolkit: if you're feeling a bit sniffly, you can still wear a mask. We all have masks, we all got really good at wearing them, and it's a courtesy to your classmates to take this easy step to decrease the likelihood of spreading anything. (Including colds and other stuff! Masks help us not spread *lots* of things.)

What if the professor gets sick? Same as for students: if I'm feeling a little sniffly, I'll wear a mask, and if I am more seriously sick (but well enough to teach), I'll zoom myself into the class. If necessary I can teach from a zoom window on the projector screen (and have done so!); I'll post to the Slack and send an email with instructions as soon as I know I need to do this.

Grading scale

I tend to grade hard on individual assignments, but compensate for this in the final grades. The grading scale will be approximately as follows:

A−	[85, 90)	A	[90, 95)	A+	[95, 100]
B−	[70, 75)	B	[75, 80)	B+	[80, 85)
C−	[55, 60)	C	[60, 65)	C+	[65, 70)
D−	[40, 45)	D	[45, 50)	D+	[50, 55)

While there will be no "curve" in the statistical sense, I may slightly adjust the scale at the end of the term if it turns out some of the assignments were too difficult. Final grades of A+ are recorded as an A in the grading system. Final grades below the minimum for D− are recorded as an F.

Note that *individual* grades recorded in Canvas should be accurate (and you should let me know if there's a data entry error!), but *averages* as computed by Canvas sometimes are not, if the averaging is complex or (especially) if an individual student has a special case scenario. The reference gradebook is my own spreadsheet, and while I will try to make Canvas reflect it (including averages) as well as I can, Canvas can't always handle it.

Graded work

I figure that I have on average about 9 hours of your time every week, including class time as well as reading, practice, homework, and projects. If you find you're regularly spending substantially more time than this, please do come discuss it with me, so that we can ensure you're making the most effective use of your time.

Engagement. In lieu of a required textbook, I will be providing links to webpages and readings in a variety of books in the Stevens ACL library. Most days there will be reading questions to do as homework (group work). In general, I will only be checking these for completion, but they will fuel in-class discussion and other engagement. General engagement will be evaluated in two-week blocks—so you don't need to artificially say a thing every day—and interactions on the Slack channel count. 10%

Collected homework. Less often (weekly-ish), I will give an on-paper assignment that I actually intend to collect, when I want to give more detailed feedback. These homeworks will proceed in two rounds: in response to your first handin, I'll give feedback (but no grade); after you have revised it, I'll assign a grade (but no further feedback). Each problem will get 10 (correct or almost), 7 (important pieces correct), 4 (on-task but incorrect) or 0 points. The homeworks are group work: you can work with anyone in the class, or on your own if you prefer, and hand in one copy for the group. 10%

All homeworks are due at the beginning of class on the due date.

Projects. Most of the work in this course is in the form of three-week programming projects that will let you explore a topic in somewhat greater depth. Each of you must implement each project yourself, but they are collaborative (you can talk to other people about them). 60% (15 each)

Projects will go out roughly every three weeks. Each will have two checkpoints and a followup that I'll describe in detail elsewhere. Design work is due at the start of class on its due date, and prep work and final versions are due at 8pm on their respective due dates unless I say otherwise. Followup assessments will be in class about a week after the code submission duedate.

Exams. There will be two exams, one at midterm and one for the final. Both will have a take-home portion and an in-class portion. These will be non-collaborative: **you are not permitted to discuss the exam with *anyone at all* other than me.** 20% (10 each)

Topics

Wk	M	W	
	August		
1	**25** Introduction Physical storage, B-trees **Project 1 out**	**27** 234-trees Red-black trees **Project 1 prep due**	
	September		
2	[Labor Day **no class**]	* **3** Red-black trees continued: implementation cases **Project 1 design due**	
3	**8** Red-black deletion cases	**10** Tries, Huffman coding Basics of information theory Compression Lossy vs lossless compression **Project 1 due**, **Project 2 out**	
4	**15** Probability review Conditional probability Bayes' Law	**17** Conditional independence Bayesian inference **Project 2 prep due**, **Project 1 followup**	
5	**22** Impl'ing count classifiers Using maps Maps of maps **Project 2 design due**	**24** Naïve Bayes Information retrieval Precision and recall	
		October	
6	**29** User interfaces UI perception and cognition Affordances, feedback Diversity and accessibility	**1** Paper prototyping **Project 2 due**	**
7	**6** Design tradeoffs UI evaluation criteria and standards **Exam 1 TH out**	**8** — **Exam 1**, **Project 2 followup**	

* **2 September**: Deadline to add/drop classes (5pm)

Wk	M	W
	October	
8	**13** Graph representations Pathfinding problems Brute-force: DFS, BFS, IDDFS	**15** Dijkstra's algorithm Using priority queues **Project 3 out**
9	**20** Heuristics part 1: A and A*, Admissibility Implementing best-first search Using hash tables	**22** Good hash functions Stateful comparators **Project 3 prep due**
10	**27** Project design **Project 3 design due**	**29** Problem spaces Game playing as search Backtracking Minimax
	November	
11	**3** Heuristics part 2: Alpha-beta pruning What is intelligence? Optimality vs emulation Knowledge vs reasoning	**5** * Database components Relational models Schemas **Project 3 due**, **Project 4 out**
12	**10** SQL	**12** SQL cont'd Entity-relationship models E-R diagrams **Project 4 prep due**, **Project 3 followup**
13	**17** Database design work **Project 4 design due**	[Research Day **no class**]
14	**24** DB constraints Converting between E-R, relational models DB design principles DB correctness: ACID	[Thanksgiving **no class**]
	December	
15	**1** Computational geometry Convex hulls Algorithm analysis **Project 4 due**	**3** Segment intersection Edge lists **Exam 2 TH out**

Exam 2 and Project 4 followup: Mon 8th, 3–5:30pm

* **5 November**: Deadline to elect pass/fail or withdraw from a class (5pm)

Policies

You can find several university-wide course policies at
`http://www.longwood.edu/academicaffairs/syllabus-statements/` .

Systems and environments

The supported systems for all programming assignments are the department machines in the Stevens ACL lab. You are generally welcome to use any systems you like for development, but it is your responsibility to transfer your work to those machines *and ensure that they work* on those systems before handing in.

Attendance and late policy

Attendance is required, and assignments must be turned in on time. That said, if you have a good reason to miss class or hand something in late, I tend to be fairly liberal with extensions if you ask in advance. (Good reasons do include assignments due for other classes.) (And medical and family emergencies are exempted from the "in advance" part, of course. But contact me ASAP.)

Frequent absence will result in a lowered participation grade; habitual absence may in extreme cases result in a failing grade for the class. *Unexcused* late assignments will normally be given a zero.

Honor code policy

Above all, I ask and expect that you will conduct yourself with honesty and integrity—and not to ignore the other ten points of the Honor Code, either. Take pride in what you are capable of, and have the humility to give credit where it is due.

The two main forms of academic dishonesty are "cheating" and "plagiarism". "Cheating" is getting help from someplace you shouldn't, and "plagiarism" is presenting someone else's idea as if it's your own. If you ever find yourself inclined towards either of these, know that there are always other, better options. Persevere! See my website[1] for some discussion and examples of

[1] `http://cs.longwood.edu/~dblaheta/collab.html`

how to steer clear of these problems, and feel free to come talk to me if you need help finding some of those other options (even if it's for another course).

Cheating or plagiarism (on any assignment) will normally receive a *minimum* penalty of lowering the *course* grade by a full letter, and may range at my discretion up to an F *in the course*. Cases will also be turned in to the Honor Board. But: I believe in your potential, and I hope that you will, or will grow to, observe this policy not simply to evade punishment but positively as a matter of character.

Accommodations

If you have any special need that I can accommodate, I'm happy to do so; come speak to me early in the term so we can set things up. If you have a documented disability, you should also contact Longwood's Accessibility Resources Office (Brock Hall, x2391) to discuss some of the support the college can offer you. All such conversations are confidential.

Inclement weather policy

I don't plan to cancel class for weather unless the entire college shuts down; and if the campus closes, I'm likely to hold class in some form by zoom instead (check your email). If you are commuting or are otherwise significantly affected by a weather event, use your own best judgement (and remember that zoom is an option); and if you do miss class for this reason (e.g.: power's out too), contact me as soon as possible to make up missed work.

Early bird policy

Nobody's perfect, and on occasion an assignment gets written a little unclearly (or, once in a while, with an actual error in it). If you catch one and bring it to my attention early, so that I can issue a clarification or correction to the rest of the class, there'll be some extra credit in it for you.

This document was written and prepared without the use of generative AI.