

Syllabus

CMSC 162: Intro to algorithmic design II

Fall 2023

Lecture: MWF 10, Rotunda G54
Lab: T 11, Rotunda G56
Websites: <https://canvas.longwood.edu/courses/1309966>
<http://cs.longwood.edu/courses/cmsc162>

A continuation of CMSC 160. Topics include algorithmic design, complexity analysis, abstract data types, and encapsulation and basic data structures. Advanced topics using a modern high-level programming language such as inheritance, overloading, and use of objects. Prerequisite: Grade of C- or better in CMSC 160. 4 credits.

Professor: Don Blaheta
Office: Rotunda 334
Phone: x2191
Email: blahetadp@longwood.edu
100% office hours: Mondays 2–3:30pm; Wednesdays 1–2pm;
Thursdays 1–2pm; Fridays 11am–noon; (see note below)

Overview

You have by now acquired some basic skills of programming and analysis, but the programs you've written have (of necessity) been small and the data uncomplicated. In this course you will continue to develop your programming skills, but more importantly, you will learn how to build layers of abstraction (and use abstractions that others have built) that will enable you to write and understand larger and more interesting programs and processes.

Textbook and resources

The main book for this class is *CS2 Software Design & Data Structures* by the OpenDSA project. It is free and online.¹

¹<https://opensa-server.cs.vt.edu/OpenDSA/Books/CS2/html/>

Some of the later readings come from its partner book *CS3 Data Structures & Algorithms* by the same authors. Also free and online.²

The other main resource is provided by us: 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-162`).

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), 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.

(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.)

²<https://opendsa-server.cs.vt.edu/OpenDSA/Books/CS3/html/>

Covid-19 notes

This section is happily much-abridged from the version I wrote in the first year(s) of the pandemic, but some attention to Covid-19 is still relevant.

Attending class. There are two ways you can attend class: in person, or via Zoom link. Either mode of attendance is sufficient for purposes of evaluating your presence and participation; if you attend via Zoom link,

- 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 covid-related I'm not going to police that. (Basically: 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, and it's definitely harder to participate fully when remote. 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 from "pandemic" to "endemic" on Covid-19, I'd just like to remind everyone that masking is still a tool in our toolkit. If you have had a Covid-19 exposure, or even just feel a bit sniffly today, you're not required to zoom (and, as noted, we do prefer in-person attendance where possible) but I do encourage you to 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 send an email with instructions as soon as I know I need to do this.

Course outcomes

At the end of this course, the successful student will be able to:

- identify appropriate implementations for abstract data types such as stacks, queues, lists, sets, trees, and maps;

- explain, implement, and use data structures such as linked lists, trees, and hash tables;
- compare and contrast standard algorithms using complexity analysis; and
- apply object-based principles to creating understandable and maintainable solutions to problems.

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.

Special note re mastery lab: You must eventually complete the first lab satisfactorily in order to get higher than a D+ for the course. See details below.

Content

Calendar

Wk	M	T	W	F
	August			
1	21 — Introductions Policies	22 — Lab 1: Review and mastery	23 §1.1 What is a Data Structure? Design and specification	25 §§2.1–2.1.1.1 Object-Oriented Design Classes and methods
2	28* §2.2 .h files Templates UML	29 — Lab 2: Classes, I/O, 2D arrays	30 — Class design cont'd	September
3	[Labor Day] no class	5 — Lab 3: Function design Unit testing	6 §§3.2–3.2.1 Implementing an ADT	1 §§1.2, 3.1 ADTs Lists
4	11 — Pointers, cont'd	12 — Lab 4: Pointers	13 — Dynamic allocation	8 §§3.2.2, TBA append, remove Pointers “Smart” pointers
5	18 §10.1 Linked List	19 — Lab 5: Linked node methods	20 — Linked List implementation, ctd	15 §§7.1–7.2 Recursion Fibonacci Linked nodes
6	25 TBA Binary search The call stack	26 — Lab 6: Reading code make, gdb Backtracking	27 — Recursive backtracking Exam 1 TH out	22 §7.7 Tower of Hanoi
	October			
7	2 §6.1 Stacks and recursion Array-based stacks Exceptions	3 — Lab 7: Using STL stack	4 — Review allocation, references, memory models	[Fall Break] no class

* **28 August:** Deadline to add/drop classes (5pm)

** **29 September:** Deadline to elect pass/fail option (5pm)

Wk	M	T	W	F
	October			
8	9 — Classic ADTs The “big picture”	10 — Lab 8: Empirical efficiency	11 §§4.2, 4.5 Algorithmic efficiency Big-O notation	[no class]
9	16 §10.2 Comparing implementations Linked Stacks Array List, Linked List revisited	17 — Lab 9: Interfaces and multiple implementations	18 Ch. 8 Quadratic sorts	20 CS3 §§8.9–8.10 Faster sorts: mergesort comparing alg’s
10	23 CS3 §8.11 Faster sorts: quicksort	24 — Lab 10: Overloading operators	25 §§9.1.1, 9.2 Queues Linked Queue	27 §§11.1–11.3 Trees Traversals
11	30 CS3 §7.8 Tree implementation	31 — Lab 11: Linked trees	November	
			1 * — Tree implementation, ctd	3 §§11.4–11.4.2 Binary search trees
12	6 §11.4.3 BST remove	7 — Lab 12: BST implementation	8 §11.4.4 BST analysis, balance, rotation	10 CS3 §§6.4, 7.12 Maps/Dictionaries
13	13 CS3 §§10.1–10.4 Hash tables	[Symposium Day no class]	15 CS3 §7.17 Heaps	17 §2.1 Inheritance is-a / has-a Hierarchies
14	20 — Model presentation Presentation debrief	21 — Lab: DT/Alg implementation	[Thanksgiving no class]	[Thanksgiving no class]
15	27 — Presentation work day	28 — Lab: DT/Alg implementation	29 — Presentations	December 1 — Presentations Exam 2 TH out

Exam 2: Thu 7th, 8–10:30am

* **1 November:** Deadline to withdraw from a class (5pm)

Grading breakdown

I figure that I have on average about 12 hours of your time every week, including class and lab 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. The work you do for this course will be evaluated as follows:

Engagement. You need to be actively engaged in this class. Engagement comes in many forms, but I expect that you will be interacting with your classmates, and with me, in class. 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. Occasional reading quizzes will also count in this category. Engagement makes up 5% of the course grade.

Labs and homework. An important part of learning happens when you try things outside of the classroom, i.e. home-work. In this course, it comes in two flavours: programming work, which will generally be connected to our once-a-week lab sessions and last about a week per assignment; and theoretical work, which will generally be due after just a few days but you'll have a chance to revise it. Programming work should be done basically on your own, but within limits you can talk to your classmates about it. (I call this work “collaborative” and go into much detail in my collaboration policy.) Theoretical homework will be group work, and you can hand in one copy for the whole group.

Labs and homework will be collectively worth 45% of the final grade.

Mastery lab. The first lab is special in that it covers no new material and is a review of programming skills you should have from CMSC 160. The regular handin deadline (after one week) will only be for full credit; you will be able to continue working on the lab problems until you get them correct. In fact, you will need to: to pass this course you must (at least eventually) correctly complete at least ten of the fourteen problems.

The highest grade you can receive in this course, without completing at least 10/14 of the Lab 1 problems, is a D+, regardless of any other assignment grades you receive in the course.

The points for the lab will otherwise count normally as a lab grade.

Presentation. At the end of the term, you'll give a presentation about a data structure or algorithm not otherwise covered in the course. This will be 10% of your grade.

Exams. There will be two exams, one in late September and one during the finals period. Each will have a take-home component and a sit-down portion. The final will not be explicitly cumulative, though of course the material from the second half of the course builds on the earlier stuff. **You are not permitted to discuss the exams, *at all*, with anyone other than me.** Each exam is worth 20% of the grade.

Policies

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

Support

This is an introductory course. That means that what is covered is an important basis for other work in the field, *not* that it is supposed to be obvious, or easy. So don't feel bad if something doesn't click right away. Never hesitate to ask me a question; I'll usually at least give you a hint as to where to look next.

“Office hours”

If I'm in my office and my door is open, that means I'm available for you to drop in and ask questions, and I'm happy to turn on my “office hours” zoom link so you can join me that way instead. At least four hours a week I've designated as 100% hours, i.e. there's a nearly 100% chance I'm available at those times.

But I'm in my office a lot and I'd like to effectively communicate to you when you're most likely to catch me, so if you look at my office schedule on my website or linked from Canvas, you'll also see many hours listed with other percentages like 60% or 40 or 10, as informal estimates of the probability I'll have office hours in that block for drop-in questions. (If you want more certainty, you can always give me advance notice and be extra sure I'll be there at whatever time!)

If you can't catch me in my office, email or Slack is probably your best bet.

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 website³ for some discussion and examples of 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.

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

³<http://www.cs.longwood.edu/~dblaheta/collab.html>

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.

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.