Syllabus
CMSC 162: Intro to algorithmic design II

Fall 2021

Lecture:	MWF 10, Rotunda 352
Lab group 1:	Mon 4pm, via Zoom
Lab group 2:	Tue 11am, via Zoom
Websites:	`https://canvas.longwood.edu/courses/1301941`
	`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:	Tuesdays 10–11am; Wednesdays 3-4:30pm;
	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://opendsa-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 be expected to have a device (your computer, or a phone or tablet) that is capable of recording and playing audio and video, and connecting to a live meeting via Zoom.

You will be expected to have reasonable bandwidth to connect to meetings, work on assignments, and occasionally upload video, *at your home* or wherever you plan to go in the event the campus closes down. If this is likely to be a problem, contact me early to see if we will be able to work around it.

You will be expected to have (and wear) a mask or the equivalent.

Covid-19 notes

There are a number of policies specific to running a class in a pandemic that I wanted to put early in the syllabus to get your attention.

Attending class. There are two ways you can attend class: in person, or via Zoom link. Either mode of attendance is equivalent 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.) Unlike last year, I can't promise that the Zoom experience will be anywhere near equivalent to the in-person experience; in fact, it almost certainly won't be. But if you are quarantined, or otherwise just can't attend in person on a particular day, zooming is better than total absence.

If you are on Zoom on a day that you present something, I *will* expect you to connect with video, at least for that part, unless you have contacted me *in advance*.

[2] `https://opendsa-server.cs.vt.edu/OpenDSA/Books/CS3/html/`

Important note: I will not necessarily turn on the Zoom link for every class day unless I know for sure someone will be attending that way. Try to get word to me that you'll need to zoom as soon as you know.

Medical needs. There are a number of medical reasons why attending class in person may not be appropriate for you. Obviously, if you receive a positive Covid-19 test, you will need to remain in isolation and attend class via Zoom link. Even without a positive confirmation, if you are feeling some symptoms or have been exposed or are awaiting test results, attending via Zoom link from quarantine is most appropriate (*especially* if you are unvaccinated).

> **More serious medical needs.** If you are feeling serious symptoms of Covid-19 (or some other sickness), your priority should be on dealing with that. If you end up missing class sessions and/or assignments due to being sick, notify me when you can and then let me know when you're on the upswing so we can plan out how to get you caught up.

Wearing a mask. As of the start of the semester, all students attending class in person must be wearing a mask or other appropriate face covering. Coverings that are acceptable include some kinds of folded bandannas, gaiters, or scarves, as long as: it covers both your nose and mouth, with two layers of cloth, fitted relatively snugly around the edges, and reduces aerosols (i.e. it's relatively tightly woven, not very stretchy, and doesn't have an "exhaust port"). If you show up to class without a face covering, you will be required to put one on or leave. If you are medically unable to wear a mask, please contact the Accessibility Resources Office to help you work out an accommodation.

It is possible that the university's masking rules will relax at some point. Even if/when they are not generally required, I will encourage you to wear a mask if you are unvaccinated, if you are mildly symptomatic (e.g. "it's probably allergies but just in case"), or if it makes you feel safer or more comfortable to do so.

All-online? It's still slightly possible that at some point in the semester we'll have to move all-online to handle an outbreak (or that I will land in quarantine). Should that happen, we will migrate the course to Zoom meetings but otherwise carry on. I expect that this course will remain largely synchronous (i.e. we meet at our regular class time) even if we go remote, but some calendar dates may be adjusted.

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	M/T	W	F
	August			
1	**23** — Introductions Policies	**23/24** — Lab 1: Review and mastery	**25** §1.1 What is a Data Structure? Design and specification	**27** §§2.1–2.1.1.1 Object-Oriented Design Classes and methods
			September	
2	**30** §2.2 .h files Templates UML	**30/31** * — Lab 2: Classes, I/O, 2D arrays	**1** §§1.2, 3.1 ADTs Lists	**3** §§3.2–3.2.1 Implementing an ADT
3	[Labor Day **no class**]	**[7]** — Lab 3: Function design Unit testing	**7/8** §§3.2.2 More implementation append, remove	**10** TBA Pointers "Smart" pointers
4	**13** — Pointers, cont'd	**13/14** — Lab 4: Pointers	**15** — Dynamic allocation	**17** §§6.1–6.2 Recursion Fibonacci Linked nodes
5	**20** §9.1 Linked List	**20/21** — Lab 5: Linked node methods	**22** — Linked List implementation, ctd	**24** §7.7 Tower of Hanoi Binary search
				October
6	**27** TBA The call stack	**27/28** — Lab 6: Reading code `make`, `gdb` Backtracking	**29** §5.1 Recursive backtracking Other uses of stacks	**1** ** §5.2 Allocation, references, memory models **Exam 1 TH out**
7	**4** — **Exam 1**	**4/5** — Lab 7: Using STL `stack`	**6** — Stacks and recursion Array-based stacks Exceptions	[Fall Break **no class**]

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

** **1 October**: Deadline to elect pass/fail option (5pm)

Wk	M	M/T	W	F
	October			
8	**11** — Classic ADTs The "big picture"	**11/12** — Lab 8: Empirical efficiency	**13** §§4.2, 4.5 Algorithmic efficiency Big-O notation	**15** §9.2 Comparing implementations Linked Stacks Array List, Linked List revisited
9	**18** §2.1 Inheritance is-a / has-a Hierarchies	**18/19** — Lab 9: Interfaces and multiple implementations	**20** Ch. 7 Quadratic sorts	**22** CS3 §§8.9–8.10 Faster sorts comparing alg's
10	**25** CS3 §8.11 Faster sorts, ctd	**25/26** — Lab 10: Overloading operators	**27** §§8.1.1, 8.2 Queues Linked Queue	**29** §§10.1–10.3 Trees Traversals
	November			
11	**1** CS3 §7.8 Tree implementation	**1/2** — Lab 11: Linked trees	**3** * — Tree implementation, ctd	**5** §§10.4–10.4.2 Binary search trees
12	**8** §10.4.3 BST remove	**8/9** — Lab 12: BST implementation	**10** CS3 §§6.4, 7.12 Maps/Dictionaries	**12** §10.4.4 BST analysis, balance, rotation
13	**15** CS3 §7.17 Heaps	[Symposium Day **no class**]	**17** CS3 §§10.1–10.4 Hash tables	**19** — Model presentation Presentation debrief
14	**22** — Presentation work day	**22/23** — Lab: DT/Alg implementation	[Thanksgiving **no class**]	[Thanksgiving **no class**]
			December	
15	**29** — Presentations	**29/30** — Lab: DT/Alg implementation	**1** — Presentations	**3** — Presentations **Exam 2 TH out**

Exam 2: Thu 9th, 8–10:30am

* **3 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 it's ok if most of your engagement is via the discussion boards as long as *some* of it is spoken out loud. In addition, there will be occasional required interactions via Canvas that will be considered part of the engagement grade. 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 early October 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 probably also have my "office hours" zoom link active, which means you're free to 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 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 website[3] 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://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. If you are commuting or are otherwise significantly affected by a weather event, use your own best judgement; and if you do miss class for this reason, 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.