Syllabus tl;dr
CMSC 162: Intro to algorithmic design II

Fall 2023

Meets:	MWF 10, Rotunda G54; and T11, Rotunda G56
Websites:	`https://canvas.longwood.edu/courses/1309966`
	`http://cs.longwood.edu/courses/cmsc162`
Professor:	Don Blaheta, Rotunda 334, `blahetadp@longwood.edu`
100% office hours:	Mondays 2–3:30pm; Wednesdays 1–2pm;
	Thursdays 1–2pm; Fridays 11am-noon

Textbook and resources

CS2 Software Design & Data Structures by the OpenDSA project.

`https://opendsa-server.cs.vt.edu/ODSA/Books/CS2/html/`

(Later, for a few readings, also its partner CS3 book)

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.

Graded work

- Engagement 5%
- Labs and homework 45%
- Presentation 10%
- Exams 20% each

  Exam 1 is out Wednesday, 27 September, in-class portion on 29 September
  Exam 2 is out Friday, 1 December, in-class portion 7 Dec @8am

Zoom attendance quick links

CLASS	OFFICE HOURS

Presentations and final project

In the last weeks of the term, each student will, with a partner or two, give a presentation about a data structure or algorithm as well as writing an implementation relevant to it. The presentation will be 12–15 minutes and needs to include:

- Accurate example diagrams
- Pseudocode and tracing using the example
- A demonstration of either correctness or efficiency

Both/all partners must participate in the presentation but may divide the time as they see fit. More details will come later in the term.

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 in the syllabus and in the Lab 1 handout.

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
				September
2	**28** * §2.2 .h files Templates UML	**29** — Lab 2: Classes, I/O, 2D arrays	**30** — Class design cont'd	**1** §§1.2, 3.1 ADTs Lists
3	[Labor Day **no class**]	**5** — Lab 3: Function design Unit testing	**6** §§3.2–3.2.1 Implementing an ADT	**8** §§3.2.2, TBA append, remove Pointers "Smart" pointers
4	**11** — Pointers, cont'd	**12** — Lab 4: Pointers	**13** — Dynamic allocation	**15** §§7.1–7.2 Recursion Fibonacci Linked nodes
5	**18** §10.1 Linked List	**19** — Lab 5: Linked node methods	**20** — Linked List implementation, ctd	**22** §7.7 Tower of Hanoi
6	**25** TBA Binary search The call stack	**26** — Lab 6: Reading code `make`, `gdb` Backtracking	**27** — Recursive backtracking **Exam 1 TH out**	**29** ** — **Exam 1**
	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
			November	
11	**30** CS3 §7.8 Tree implementation	**31** — Lab 11: Linked trees	**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**]
				December
15	**27** — Presentation work day	**28** — Lab: DT/Alg implementation	**29** — Presentations	**1** — Presentations **Exam 2 TH out**

Exam 2: Thu 7th, 8–10:30am

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