Syllabus
CMSC 262: Data structures and algorithms in application

Fall 2017

Time:	TR 2pm
Room:	Ruffner 354
Website:	`http://cs.longwood.edu/courses/cmsc262`

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.

Professor:	Don Blaheta
Office:	Ruffner 337
Phone:	x2191
Email:	`blahetadp@blahedo.org` or `blahetadp@longwood.edu`
Office hours:	Mon 2–3pm, Tue 11am–noon, Thu 1–2pm, Fri 1–2pm

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.

Objectives

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.

Books

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.

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, 100)	[1]	
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.

[1] Alas, no A+, unfortunately.

Graded work

Daily work. 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; and most days there will be a set of reading questions to do as homework (group work is fine). In general, I will only be checking these for completion, but they will fuel in-class discussion. Being present, attentive, and responsive are prerequisites for these points.

Collected homework. Less often, 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. Each problem will get 5, 3, 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.

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).

Projects will go out roughly every three weeks. Each will have two checkpoints 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 4pm on their respective due dates unless I say otherwise.

Exams. There will be two exams, one at midterm and one for the final. Both will be take-home, and you will be given a few days to work on them. These will be non-collaborative: **you are not permitted to discuss the exam with *anyone at all* other than me.**

Breakdown

Daily work	10%	
Collected homework	10%	
Projects	60%	(15 each)
Exams	20%	(10 each)

Topics

Wk		T	R	
		August		
1		**22** Introduction Physical storage, B-trees **Project 1 out**	**24** 234-trees Red-black trees **Project 1 prep due**	
2	*	**29** Red-black trees continued: implementation cases **Project 1 design due**	**31** Red-black deletion cases	
		September		
3		**5** Tries, Huffman coding Basics of information theory Compression Lossy vs lossless compression	**7** Probability review Conditional probability Bayes' Law **Project 1 due**, **Project 2 out**	
4		**12** Conditional independence Bayesian inference Naïve Bayes classifiers	**14** Information retrieval Precision and recall **Project 2 prep due**	
5		**19** Impl'ing count classifiers Using maps Maps of maps **Project 2 design due**	**21** User interfaces UI perception and cognition Affordances, feedback Diversity and accessibility	
6		**26** Paper prototyping	**28** Design tradeoffs UI evaluation criteria and standards **Project 2 due** **Exam 1 out**	**
		October		
7		**3** Graph representations Pathfinding problems Brute-force: DFS, BFS, IDDFS **Exam 1 due**	**5** Dijkstra's algorithm	

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

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

Wk		T	R
		October	
8	*	**10** Heuristics part 1: A and A* Admissibility **Project 3 out**	**12** Impl'ing best-first search Using priority queues Using hash tables
9		[Fall break **no class**]	**19** Good hash functions Stateful comparators **Project 3 prep due**
10		**24** Problem spaces Game playing as search Backtracking **Project 3 design due**	**26** Minimax Heuristics part 2: Alpha-beta pruning
			November
11		**31** What is intelligence? Optimality vs emulation Knowledge vs reasoning Theorem proving as search	**2** Database components Schemas **Project 3 due**, **Project 4 out**
12		**7** SQL	**9** Entity-relationship models E-R diagrams **Project 4 prep due**
13		**14** DB constraints Converting between E-R, relational models DB design principles **Project 4 design due**	**16** DB correctness: ACID Network DB issues Bandwidth, latency, scale Security
14		**21** Computational geometry Convex hulls	[Thanksgiving **no class**]
15		**28** Convex hulls cont'd Algorithm analysis **Project 4 due**	**30** Segment intersection Edge lists **Exam 2 out**
		December	
		Exam 2 due: Thu 7th, 2pm	

* **9 October**: Deadline to withdraw from a class (5pm)

Policies

Systems and environments

The supported systems for all programming assignments are the department machines in the Hardy House. 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[2] 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).

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

Cheating or plagiarism (on any assignment) will normally receive a *minimum* penalty of a lowered *course* grade, ranging 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 Office of Disability Resources (Graham 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. 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.