Syllabus
CMSC 208: Grammars, Languages, and Automata

Spring 2026

Time:	TR 2pm
Room:	Rotunda 356
Website:	`http://cs.longwood.edu/courses/cmsc208/`

This course introduces topics in theoretical computer science including formal languages, grammars, and computability, which form the basis for analysis of programs and computation. These tools are then used to explore several modern programming languages and survey the major programming paradigms. Prerequisite: CMSC 162, MATH 175. 3 credits.

Professor:	Don Blaheta
Office:	Rotunda 334
Phone:	x2191
Email:	`blahetadp@longwood.edu`
100% office hours:	Mondays 3–4pm; Tuesdays 1–2pm; Wednesdays 10–11am; Thursdays 11am–noon

Overview

Once you've gotten past the basics of programming and thinking computationally, it becomes important to develop the language and habit of mind that lets you speak very precisely about computation itself. In this course, we see the mathematical underpinnings of the discipline of computer science: an introduction to computational theory, illustrated in part by programming languages with (at least) four distinctly different ways of expressing computation.

Course objectives

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

1. write and interpret mathematically valid proofs, including induction proofs;
2. construct and manipulate automata and regular expressions that accept specific languages;
3. identify salient differences between various programming languages and language paradigms; and
4. explain the nature and significance of *NP*-completeness.

Book and resources

A recommended resource is John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman, *Introduction to automata theory, languages, and computation.* Either the 2nd or 3rd edition is fine (they're basically identical). Second edition: ISBN 0-201-44124-1. Third edition: ISBN 978-0-321-45536-9. There is a copy of the 2nd edition in the Stevens lab.

In addition, there will also be language tutorials and references posted to the course website.

For the programming assignments, you will have an account on the department machines in the Stevens lab. In some cases you may be able to do the assignments on your own computer, but the officially supported option will be the department machines.

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-208`), 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. (See also the "Covid-19 notes" section at the end of this document.)

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.)

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.

Topics

Section numbers represent readings in the Hopcroft et al book; you are welcome and encouraged to read about the topics in advance of class or afterwards (as well as other resources if you prefer).

Wk	T	R
	January	
1		**15** — Languages as syntax; grammars. Intro to LaTeX
2	**20** §1.2 Logic review Proofs	**22** * §1.3.1 Sets review; set equivalence. Functions and relations.
3	**27** §§1.3.2–1.3.4 Contradiction proofs. Intro to Racket	**29** More Racket
	February	
4	**3** §§1.4–1.4.2 Recursive definitions. Integer induction. Racket lists and recursive functions.	**5** §§1.4.3–1.4.4 Structural induction. More recursive functions.
5	**10** §1.5 Languages as sets of strings. Racket first-class functions.	**12** §§2.1–2.2.1 Finite automata.
6	**17** §§2.2.2–2.2.5 Accepting languages. Alternative DFA representations. Basics of Perl.	[Professor out **no class**]
7	**24** ** §§2.3.1–2.3.2, 2.5.1–2.5.2 Nondeterministic finite-state automata. Regular expressions and FSAs.	**26** §3.1 Regular expressions. Perl-compatible regular expressions.
	March	
8	**3** §3.3 Applications of regular expressions. Tokenising LISP-family languages.	**5** **Exam 1**

* **22 January**: Deadline to add/drop classes (5pm)

** **24 February**: Symposium Day (C410 students)

Wk	T	R
	March	
	SPRING BREAK	
9	**17** — More FA practice.	**19** §§4.1,51.–5.1.3 The pumping lemma. Context-free grammars.
10	**24** §§5.1.4–5.2.2 Grammars for Scheme, C, Perl; Backus-Naur Form. CFG derivations. Parse trees. Intro to C♯.	**26** §§5.3–5.3.1 Parsers. Parsing to abstract syntax trees. Syntax-driven semantics.
		April
11	**31** — Review language translation pipeline; interpreting vs compiling. Case-based code generation. C♯ and polymorphism.	* **2** cont'd
12	**7** §§8.1,8.2–8.2.3 Decidability. Turing machines.	**9** — Basics of Haskell.
13	**14** — Type inference in Haskell.	**16** — Static and dynamic type checking; Scheme, Perl, C♯, Haskell. Lazy evaluation and currying in Haskell.
14	**21** — Lambda calculus; Church-Turing thesis.	**23** §§9.1–9.2,10.0–10.1.2 Recursively enumerable languages. The halting problem. Review big-O notation; complexity and P.
15	**28** — Recurrence relations in analysis. The Master theorem.	**30** §§10.1.3–10.2.1 Verifiability, NP, and NP-completeness.
	May	
	Exam 2: Tuesday 5 May, 3–5:30pm	

* **1 April**: Deadline to withdraw from classes or declare P/F (5pm)

Graded work

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, as will intermittent checks as to whether you've attempted the homework. Engagement makes up 5% of the course grade.

Theoretical homework. One of the major strands of this class is an introduction to theory of computation, and it's one of the areas of computer science that most needs practice (and collaboration) to really wrap your head around. On a roughly weekly basis I'll assign homework to practice all this; theoretical homework is group work and has a revision cycle, and collectively makes up 15% of the final grade.

Practical homework. We will be touching on four different programming languages that will (probably) be new to you, and you will have to write short programs in them. As always in one of my courses, start *early* on these so you can come for help if you get stuck! You can collaborate a bit (i.e. talk to people) but everyone will hand in their own work and shouldn't be copying anyone else's. The practical stuff makes up 20% of the final grade.

Exams. There will be two exams: one at midterm and one for the final. My current plan is for each to have a take-home component that you'll have two or three days to work on, and which you'll bring with you to the in-class portion of the exam. Each exam is worth 30% of the grade.

Breakdown

Theoretical homework	15%	
Practical homework	20%	
Exams	60%	(30 each)
Engagement	5%	

Policies

Support

I'm in my office a lot (not just during posted office hours). Feel free to come in and ask questions (or just to talk). 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[1] 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.

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

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.

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.

The section formerly known as "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 link. Either mode of attendance is sufficient to mark you as "attending" (not necessarily engaging or participating). 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. (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 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'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.

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