CS 13: Mathematical Foundations of Computing — Spring 2023

Course Syllabus

Information At-A-Glance

Instructor	
Name:	**Adam Blank**
E-mail:	blank@caltech.edu
Office:	ANB 115
Office Hours:	Mon: TBD Thu: TBD Fri: TBD Or by private meeting.

Course Website
https://math4cs.com
Visit early. Visit often.

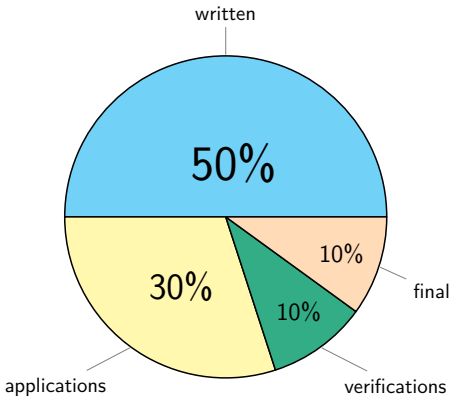
Lecture
ANB 105 on MWF
11:00 AM – 11:55 AM

Course Overview

This course is intended to prepare students for future work in proof-based CS courses like CS 21 and CS 38. It is organized around *cool computer science applications* of raw (mostly discrete) mathematical topics. CS 1 is a co-requisite as there will be a small number of programming assignments. We will cover basic set theory, induction and inductive structures (e.g., lists/trees), asymptotic analysis, combinatorics, number theory, and graph theory. Applications will include number representation, basic cryptography (RSA), basic algorithms on trees, numbers, and polynomials, a bit of graphs as social networks, compression and (simple) error-correcting codes.

Assessments

This course will consist of four types of assessments: written proof-based problem sets, programming projects, verification sets, and exams. As this is the first time this course is running, we reserve the right to change the percentages and/or delete assignments if we find that we're asking too much of the students.



Grade Cutoffs

Please note that there is no way to receive a D in this course. Any score at or below 69% is considered an F, 69-70 is a D+, 70-80 is some kind of C, 80-90 is some kind of B, and 90-100 is some kind of A.

Written Sets

Unlike many of Adam's other courses, this one will have more "traditional" written problem sets. The goal here is help prepare you for solving more difficult computer science problems in future courses as well as improve your proof-writing skills.

Programming Projects

There will be (at most) three programming projects in this course. These will be small-ish projects designed to show off how the theory we develop in lecture and on the written sets can be used in practice.

Verification Sets

More on these later.

Exams

The midterm and final exams will be **cumulative** with an equal emphasis on all the material in the course to that point. They will be entirely written, proof-based questions. The time limit will be *13 hours* though we expect nobody will use anywhere near that much time.

Late Policy

In this course, you have the opportunity to *earn* up to 10 "late tokens". Each late token will allow you to submit a problem set of any type (not including exams) up to 24 hours late; tokens are indivisible and you may not go into "token debt".

Attending a lecture will net you $\frac{1}{3}$ of a late token. You may not use partial tokens, however.

You do not need to use tokens for serious medical (physical or mental) or emotional circumstances; in such situations, contact the instructor to work out a plan for completing the work in a reasonable time frame.

Getting Help

Please don't be afraid to ask for help if you don't understand something. Adam holds *at least three* office hours a week, and they get lonely and bored if you don't show up! They also show up early to lecture and are happy to answer any questions you might have before or after lecture.

At office hours, you can ask for clarification on a lecture (or for a *repetition* of the lecture!). You can ask for help with a frustrating part of the homework. You can even show up just to tell us you're frustrated and vent.

Here's some first steps on how to get help:

- Ask on Edstem
- Ask someone on course staff questions before/after lecture, etc.
- Come to office hours

Collaboration & Academic Integrity

See our "collaboration table" on the website. We reserve the right to modify or clarify this policy as needed. Notably, you may not, under any circumstances, look at another student's/group's code.