MATH 113 Introduction to Abstract Algebra: Syllabus

Instructor:	Gabriel Dorfsman-Hopkins (gabrieldh@berkeley.edu)
Homework Grader:	Onyebuchi Ekenta (oekenta@berkeley.edu)
Lecture:	T-Th 5:00-6:30 PM (Prerecorded on Zoom)
Office Hours :	During scheduled class hours. More times TBA or by appointment
Text:	Introduction to Mathematical Cryptography, 2014 edition by Hoffstein, Pipher, Silverman [HPS] **Digital copies available through the UC Berkeley Library**
Course Website:	`http://www.gabrieldorfsmanhopkins.edu/m116f20/index.html` Grades will be posted on bcourses.

Objectives

Cryptography is the art of transmitting confidential information over a potentially monitored channel. Codes and ciphers are ancient tools, but in the modern digital age entirely new cryptographic paradigms became possible, and necessary. One of these paradigms, and the central object of study in this course, is *public key cryptography*, allowing the exchange of confidential information between two people who have never met! At the core of public key cryptography are marvelously clever and deep mathematical insights existing at the intersection of abstract algebra, number theory, and geometry, which we will introduce and explore.

This course will strike a balance between theory and practice. We will have to develop some mathematics, and along with that there will be some proof writing and abstract formulations. With these tools in hand we will be able to fully implement many of the modern cryptosystems we study, and start sending eachother confidential messages in a public forum!

Structure

Due to the ongoing COVID-19 pandemic course will be entirely remote. I will post 2 recorded lectures a week, corresponding to our scheduled Tuesday-Thursday schedule, but they will be available to watch at your convenience. I will record them on Zoom and post a link on the course website, as well as posting them to BCourses.

That being said, mathematics is not a spectator sport. Watching me do a algebraic computation may not be as useful as working it out yourself, and this is even more true for writing lines of code. For this reason I will have exercises embedded in the lectures where I will suggest you pause the video and work out some of the details.

I will also be available hosting zoom office hours during the usual course hours (5-6:30 on Tuesdays and Thursdays), so if you are watching at that time feel free to drop in and ask any questions.

Cocalc

Each one of you will have a shared folder with me on an online collaborative coding platform called CoCalc, which in particular runs SageMath, an open source mathematical software we will

be making heavy use of this quarter. It is here where you will do and turn in the computational parts of your homework assignments. I would also recommend having a Jupyter notebook open in CoCalc during the lectures to be able to follow along and work out the computational in class exercises. During the first lecture I will walk you through setting this up.

Homework

The main portion of your grade will be homework assignments. There will be homework collected almost every week **on Tuesdays**. They will be assigned the week before they are due, and will tend to come in 2 parts.

- A *written part*, which you will turn in on *Gradescope*. This can be typed up in LaTeXor hand written and scanned (please don't sumbit grainy photos). This will generally be more mathematical, involving proof writing or explicit computation.
- A *implementation part*, which will be turned in in the form of Jupyter notebook on CoCalc (for example, Homework 0 will be turned in by having Homework0.ipynb file existing in your CoCalc project folder and containing the completed assignment). These will generally be implementations of algorithms, or more elaborate computations. Please clearly mark which part of the notebook is which problem with comment lines.

Feel free to discuss homework in groups, but know that each one of you should be writing up your own assignment (no copying and pasting code will be allowed!).

Implementation Projects

There will be 2 larger implementation projects. These will look a lot like the implementation sections of your homework assignments, but will likely be larger more cohesive implementations of an entire we've been studying (rather than just single algorithms). You are allowed to use algorithms written in homework assignments as peices of thse projects. You may also use the course text, as well as your course notes. You may *not* use the internet or your peers.

Warning! As the projects are specific implementations of things we are studying in class, these due dates may shift depending on course pace. I will give you as much notice as possible if there is any change.

Project 1: Due Tuesday 10/6 in CoCalc.
Project 2: Due Tuesday 11/24, in CoCalc

Final Exam

There will be 1 final exam, which will be a takehome assignment. It will look similar to the written portions of the homework assignments, and will have both a written portion and an implementation portion. You may use the course text as well as your course notes and previous graded homework assignments. You may *not* use the internet or your peers.
Final: Assigned Thursday 12/10 Due Thursday 12/17 On Gradescope **AND** CoCalc

Grading

Raw grades will be computed* as follows:

Category	Percentage
Homework	50%
Project 1	15%
Project 2	15%
Final	20%
Total:	100%

*After evaluating the performance of the class over the entire quarter, I will adjust the median grade accordingly.

Make-ups and Extensions

If you need an extension on homework or the projects, let me know *as soon as possible*. I like to post solutions ASAP after assignments are turned in, and once they are posted I will no longer accept late assignments. I do understand that we are in the midst of a global pandemic, and things are unpredictable, so communicate with me early and often if you need more flexibility. I will be more flexible with homework assignments than I will with the projects or final.

Disabled Students' Program (DSP)

The University of California is committed to providing access, equal opportunity and reasonable accommodation in its services, programs, activities, education and employment for individuals with disabilities. These resources include exam proctoring and accommodations in distraction free environments and with extra time as well as note taking. To request disability accommodation contact the DSP Office at least ten days in advance at (510) 642-0518(V), (510) 642-6376(TTY), (510) 643-9686(FAX), or dsp@berkeley.edu.

If you have a letter from the Disabled Students Program (DSP) indicating that you have a disability which requires academic accommodations, please present the letter to me **as soon as possible** so we can discuss the accommodations you need.

COVID-19 Addendum

We are in the midst of a global pandemic, and everything is unpredictable. As such, things might change rapidly, both in the stucture of the course and how we respond to it, and we must be ready to adapt. I have a long term sub lined up in case I get sick, or have a family member I must take care of. If you or your family were to become ill don't hestitate to contact me and we will work something out to keep you from falling too behind, or being penalized. It is most important to stay safe and healthy and prevent the spread.

Course Schedule

Below is a very rough schedule of the course, organized weekly. It is very difficult to predict pace for an online course during a pandemic. We may go faster than is laid out over the first few weeks...or slower. We may cover all these things and more, or we may skip some sections due to time. Since projects are specific implementations, their due dates may shift due to timing issues.

Week 0 (8/27):	Introduction. Setting up CoCalc. Course Goals. An Example of a Cipher.
Week 1 (9/1,9/3):	Cryptanalysis. Elementary number theory. 1.1-1.4
	Homework 0 due Tuesday.
Week 2 (9/8,9/10):	More number theory. Encoding Schemes. 1.4-1.7
	Homework 1 due Tuesday.
Week 3 (9/15,9/17):	Public Key Cryptosystems. The Discrete Log Problem. 2.1-2.4
	Homework 2 due Tuesday.
Week 4 (9/22,9/24):	Group theory. Cryptanalysis of Discrete Log Problem. 2.5-2.7
	Homework 3 due Tuesday
Week 5 (9/29,10/1):	Chinese Remainder Theorem. Discrete Log Attacks. Rings. 2.8-2.10
	Homework 4 due Tuesday
Week 6 (10/6,10/8):	Eulers Formula. RSA and Implementation. Primality Testing. 3.1-3.4
	Homework 5 due Tuesday
Week 7 (10/13,10/15):	Factorization Techniques and Attacking RSA 3.5-3.8
	PROJECT 1 DUE TUESDAY
Week 8 (10/20,10/22):	Quadratic Reciprocity and Probabilistic Ecryption. 3.8-3.10
	Homework 6 Due Tuesday
Week 9 (3/23,3/27):	Pragmatic application of RSA: Digital Signatures. 4.1-4.3
	Homework 7 Due Tuesday
Week 10 (11/3,11/5):	Probabilistic Attacks. 5.3-5.5
	Homework 8 Due Tuesday
Week 11 (11/10,11/12):	Elliptic Curves. 6.1-6.3
	Homework 9 due Tuesday
Week 12 (11/17,11/19):	Elliptic Curve Discrete Log Problem. Revisiting Diffie Helman. 6.4-6.6
	Homework 10 due Tuesday.
Week 13 (11/24):	*No Class Thursday* Elliptic Curves in Characteristic 2. 6.7
	PROJECT 2 DUE TUESDAY
Week 14 (12/1,12/3):	Weil Pairings. 6.8-6.10
	Homework 11 Due Tuesday
	Last day of formal instruction on Thursday
Week 15 (12/7-12/11):	*Review Week*
	Homework 12 due Tuesday
Week 16 (12/14-12/17):	**Final Exam Due Thursday**.