Math 161:
Introduction to formal verification of mathematics

Instructor:	Philip Matchett Wood office: Science Center 514 email: `pmwood@math.harvard.edu` office hours: Monday 2–3pm, Wednesday 10–11am.
Class time:	Monday and Wednesday noon-1:15pm, location Science Ctr 112.
Textbook:	Mathematics in Lean (online), by Jeremy Avigad, Kevin Buzzard, Robert Y. Lewis, Patrick Massot.
Canvas page:	`https://canvas.harvard.edu/courses/118828`
Github:	We will use Github to submit homework online. No prior Github experience is necessary.

Course description. How can you be really sure that a mathematics proof is completely correct or that a computer program does exactly what is supposed to do? One way is with automated theorem provers, one of which is Lean, which are computer languages that can encode proofs.

Theorem provers are sometimes used in industry for hardware and software verification. For example, a large company is currently using a theorem prover to prove that two-step verification information cannot accidentally be used for advertising. Theorem provers can also formally and precisely check a proof of a statement in pure mathematics. For example, the Liquid Tensor Experiment used Lean to formally verify the main theorem of liquid vector spaces, an important and complicated theorem proved on paper by Dustin Clausen and Peter Scholze.

The goal of Math 161 is to learn how to implement proofs of mathematics statements in the Lean Theorem Prover. This course is for students who already have a background in writing proofs from at least two different courses. Some experience with programming is likely useful, but not required. You can get a quick introduction to what Lean is and the kinds of things course will be about by trying the Natural Numbers Game, where you can practice implementing proofs of some very basic facts about the natural numbers.

In Math 161, we will work through parts of the book Mathematics in Lean to build a foundation in implementing mathematics in Lean. The course will also include 2 projects—a midterm project implementing something you learned in your first or second proof-base course, and a final project implementing something you learned or are interested in this year. Each project will consist of Lean code along with a write-up explaining the code. The write-up for the final project should be between 5 and 8 pages so that it satisfies the Junior Paper requirement for math concentrators.

Prerequisites. Two proof-based courses prior to taking Math 161. You should also have a laptop that is capable of running Lean that you can bring to class.

Grades. Grades will be assigned based on the following proportions:

Weekly homework: 35%	Midterm project: 20%
Class Participation: 10%	Final Project: 35 %

Respect. Be sure to respect your fellow students and the comments they make in class. The classroom is a safe space where we all value and support the contributions of every community member.

Weekly Homework. Homework will be due each week on Wednesday at 10pm, submitted electronically via Github. (No prior Github experience is necessary—we will learn the basics as part of the class.)

Be neat when writing up homework code and include comments to explain what you are doing. Your goal is twofold: create correctly compiling code that proves the given mathematical statements, and keep the code organized and neat so that a reader can understand what the code is doing.

Academic integrity is a serious issue. The code that you turn in must be your own. You are encouraged to work out ideas in groups and talk to other students as you are working, but you must write up your own code.

Class Participation. Be there and be ready! Class will include active discussions and I will randomly prompt students to brainstorm ideas, notice something mathematical, answer questions and/or give ideas. I want to hear from everyone. We will also spend a good portion of class time coding in Lean, so be ready to get to work!

The two projects.

The midterm project should be something along the lines of implementing a proof of a mathematical statement that you learned in your first or second proof-based course. The final submission is due March 31st and must include the code you created along with a write-up explaining what you did.

The final project, which will take the place of a final exam, should focus on implementing a mathematical statement that you learned this year, or a statement with a similar mathematical level. We can discuss suitable topics for the final project as the semester gets underway. The final project and accompanying write-up will be due at the end of the final exam block assigned to this class (date TBA). The written portion of the project should be 5-8 pages and can be used to satisfy the junior paper requirement for mathematics concentrators.

Resources for Lean 3

The Lean 3 Community homepage provides instructions for downloading and installing Lean. We will work with Lean 3, though there is ongoing work towards an update called Lean 4.

Theorem Proving in Lean is a book that gives a detailed description of how Lean works and the technical aspects of how it verifies that theorems are correct.

The Lean Zulip chat is an online community of people working with and interested in Lean.