MAT 250: Introduction to Advanced Mathematics

Summer II, 2020

Instructor:	Taras Kolomatski	**Time:**	MWTh 1:30 - 4:45
Email:	taras.kolomatski@stonybrook.edu	**Place:**	Online
Website:	`http://www.math.stonybrook.edu/~tkolomat/mat250-sum20/`		

Course Overview: *Foremost, this is a course about understanding, developing, and writing proofs.* Its objective is to sharpen your critical reasoning skills and to cure you of any latent fears that you may have of abstraction. To this end, we will study many different *formal systems*—each will be logical microcosm defined by its own rules. (This breaks from the tradition in previous offerings of studying primarily number theory.)

For example, we will begin with a model of the natural numbers: a `Nat` is either the symbol `Z` (for zero), or a string of characters of the form `(S Nat)` where `Nat` is a placeholder for some `Nat`. Thus the strings `Z`, `(S Z)`, and `(S (S Z))` are members of `Nat`, while the strings `2`, `(S S)`, and `(S (Z))` are not. We will then give several definitions for arithmetic operations on `Nat`s, giving meaning to expressions such as `(+ (S Z) (S Z))`. The flavour of these definitions will be *computational*, and they may be thought of as '*programs*' defining (at a very high level) the operations that a computer would perform to evaluate certain expressions. This will all be discussed in much greater detail, but the point is that we will have a very concrete universe obeying very strict rules, and within this universe, we can ask questions of the form: *If A and B are arbitrary `Nats`, is `(+ A B) = (+ B A)`?* This will be proven from a combination of general techniques and applications of the rules defining the system.

Following this, we move on: *more systems, more rules.* Our second system will consist of expressions of the form `(if (if T F T) T (if T (if F T F) F))`, and we will discuss ways in which to endow such systems with *meaning*, and (as you can guess by now) we *develop, state, and prove* things about these constructions. Then there will be a third, fourth, fifth, *etc.* new system. Memorising the rules of each system is unnecessary in principal (and they will be given on exams), but the skill of manipulating new rules and the general proof techniques will be engraved in your brains from ceaseless repetition. You will think inside, outside, around, and behind the systems.

And then, our final system: **logic itself**! The proofs prior to this point will have been *informal*, their computational content was their making sense to your intellect. In this final twist, we will *formalise proof.* This treatment of logic will decouple the notions of *truth* and *proof.* Discussing semantics at this level will be simultaneously close to abstract human intuition and inflexible programming, and it is at this point that we can think back to how we reasoned about systems thorough the course, and how we reason on a daily basis, and wonder…

Prerequisites: This course is an *accelerated introduction to mathematics.* There are no specific formal requisites, and, in particular, I will not assume that you know how to write proofs going in (that's the point of the class). However, I will assume that you are *motivated and enthusiastic about learning.*

Relative to MAT 200, this class will demand more from you. First, the assigned work will take more you your time. Second, I will expect a greater degree of independence: you will be expected to do everything necessary to keep up with the material. This includes, emphatically, *seeking help from me if you are struggling and not giving up on submitting work mid-semester*!

`Note`: Following lectures in real time is hard - during my undergrad, I understood very little of what my professors said as they said it! Ravi Vakil, a prolific mathematician and three time IMO medallist, has a document on how to get the most out of talks. He writes: '*Talks are like horses: once you are thrown off, it is hard to get back on. Especially if the horse is stomping on your face.*' His advice is that, instead of just completely glazing over, you should try to write down three small facts (one sentence definitions or interesting remarks) that you've learned. '*If you can get even three small things out of a talk, it is a successful talk. And if you can't get even three small things out of a talk, it was not a successful experience.*'

With that being said, if you make a genuine effort to engage with the content of this course, I do not intend to give you a lower grade than you would have received in MAT 200; my position is that ***you ought to be rewarded, not punished, for having worked harder***! I will actively looking for evidence that you have made personal progress and learned at least a MAT 200 sized chunk of new material; if I am convinced that you have worked hard, I would be comfortable adjusting your grade upwards.

My intention is to provide a suitable, but not extreme, challenge. I will be vary careful to not make assumptions of prior knowledge, although the content may at times be frustrating, exhausting, or intimidating. If you show determination, it is my intention that you succeed. Ideally, this course will ideally serve as a foundation for further studies in *both mathematics and computer science.*

Resources: We won't make use of external textbooks. Lecture notes will be written and posted on the course website thorough the term. Since this is an online course, you will need a computer, stable internet connection, microphone, and camera (a smartphone camera will suffice). Since this class will involve typing up most of your work, it will not be reasonable to work solely through a cellphone or tablet. Most lectures will occur live, supplemented by some asynchronous tutorials. Recordings of lectures will be posted on the course website. Help will be available in office hours (see the course website for times), by appointment, or though the course Piazza forum.

Grading Policy: Assignments (40%), Midterm (20%), Final (40%).
The exams will primarily be in take-home format, but may potentially be supplemented by an oral component following the submission deadline. There will generally be no make-up exams; severe and documented extenuating circumstances will be handled on a case-by-case basis. If you know that you will be unable to make certain exam dates (such as for religious observance), it is your responsibility to let me know of this within the first week of class.

Assignments: Solving problems is one of the most important facets of this course. Keeping up with homework will be crucial. Exams will assume familiarity with examples introduced in assignments.

Assignments will be posted on the course website and announced in lecture at least a week prior to their deadline. **Assignment solutions are not to be handwritten**, *but must instead be typed in plaintext or typeset using LaTeX*, as indicated by the assignment. (Inserting photographs of diagrams will sometimes be permissible.) Detailed tutorials on how to do this will be provided, as well as templates of solutions, which can be modified with near-zero knowledge of LaTeX (this is what most of us do).

Some assignments will involve writing programs in the language *Racket*, which is a functional programming language descending from Scheme/Lisp. If you have any programming experience, then you will likely have to unlearn it, as this language is completely different from anything that you are likely to have previously encountered. *Racket*'s language family is unique in that, unlike the thousands of pages needed to specify a language like C++, it can be given a very brief formal specification—the language itself (restricted to the features that we will use) is a fairly uncomplicated formal system! (Indeed, as Racket is designed to be an environment for developing one's own languages, ease of formal specification is its foremost feature.)

For programming assignments, you will submit code thought Marmoset, an automated online judge that will score your problem within seconds. *You will be allowed to re-submit your code without penalty as many times as you like, prior to the deadline. Submissions made after the deadline, prior to the final exam, will be subject to a 50% penalty. Your grade on the problem is based the score of your highest submission, so making any submission (whether late or not) cannot possibly decrease your grade.* Hence, there will be no uncertainty in how programming assignments are graded. Non-programming assignments may be submitted at most 48h following the deadline, with a penalty of 30% if submitted within 24h, and of 50% otherwise.

The main point of the programming assignments is make the content of the course as concrete as possible—to give you access to an alive and ticking formal system right under your fingertips; we'll teach only a small part of an introduction to programming, and our treatment will be logically complete and self-contained. It will also tie in very closely with the content of the course; the first two examples of formal systems given in the course overview are written in the syntax of Racket.

University-wide policies:

Student Accessibility Support Center Statement: If you have a physical, psychological, medical, or learning disability that may impact your course work, please contact the Student Accessibility Support Center, 128 ECC Building, (631) 632-6748, or at sasc@stonybrook.edu. They will determine with you what accommodations are necessary and appropriate. All information and documentation is confidential.

Academic Integrity Statement: Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Faculty is required to report any suspected instances of academic dishonesty to the Academic Judiciary. Faculty in the Health Sciences Center (School of Health Technology & Management, Nursing, Social Welfare, Dental Medicine) and School of Medicine are required to follow their school-specific procedures. For more comprehensive information on academic integrity, including categories of academic dishonesty please refer to the academic judiciary website at http://www.stonybrook.edu/commcms/academic_integrity/index.html

Critical Incident Management: Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of University Community Standards any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn. Faculty in the HSC Schools and the School of Medicine are required to follow their school-specific procedures. Further information about most academic matters can be found in the Undergraduate Bulletin, the Undergraduate Class Schedule, and the Faculty-Employee Handbook.