APM 581 Theory of Computation Fall, 2011

(revised syllabus, 9/2/11)

The course: APM 581, Theory of Computation, deals with what can and cannot be computed, and how efficiently such computations can be done. In particular, we will study models of computation, such as finite state automata and Turing machines, "languages" (which provide a convenient way to talk about computational problems), and various "complexity classes" of computational problems, such as **P**, **NP**, **PSPACE**, and **BPP**. The subject matter of computational complexity is only about 40 years old. It is an exciting and active field of research, with many of the best computer scientists from government, academia, and industry engaged in it, and progress is being made every year.

Instructor: Professor Jerrold W. Grossman, 346 SEB, (248) 370-3443. My preferred e-mail address is `grossman@oakland.edu`. Rather than listing set office hours, my policy is that I'm almost always around and you are encouraged to come for help or just to chat at any time. Of course you can also make an appointment or communicate via phone or e-mail.

Prerequisites: The real prerequisite for this course is to have a fairly high level of mathematical sophistication. You will have gotten this by having taken courses in which there were definitions, theorems, and proofs. The formally listed prerequisite is a course in discrete mathematics, such as APM 263, which, in addition to providing the background in definitions, theorems, and proofs, should have given you some exposure to topics such as graphs, propositional and predicate logic, relations, strings, and big-Oh notation.

Textbook: The primary textbook for the course is *Introduction to the Theory of Computation*, second edition, by Michael Sipser (Thomson, 2006). We will probably try to proceed fairly quickly through Part One in order to concentrate on computability and complexity in Part Two and Part Three. The first edition of this book (© 1997) seems to be very similar to the second edition; the exercise sets are somewhat different, though, with some solutions included, as well as some extra exercises in the newer version. Consult other books as you wish, such as books entirely on complexity (e.g., *Computational Complexity: A Conceptual Perspective*, by Oded Goldreich (Cambridge University Press, 2008) and *Computational Complexity: A Modern Approach*, by Sanjeev Arora and Boaz Barak (Cambridge University Press, 2009), drafts of both of which can be downloaded for free from the Web—see course website).

Course website: Lots of useful information can be found here. The home page can be reached via Moodle, or you can type in the URL `https://files.oakland.edu/users/grossman/web/APM581` and bookmark it. One particularly useful link on the links subpage is to MIT's OpenCourseWare pages for their courses (graduate and undergraduate) that use Sipser's book.

Homework: Each chapter of the textbook ends with a few problems, and I will assign some of them (or other exercises that are not in the textbook) for you to solve and hand in, probably weekly. Make sure to write up your solutions well. You may (and probably should) work with other members of the class on these exercises, but you need to write up the solutions on your own.

Tests: There will be a midterm and a final exam. Details will be provided later. I expect that they will be "closed book" but "open crib sheets".

Research paper presentation: Each student will select a paper in the recent (last ten years or so) theory of computation literature (classification 68Q in MathSciNet, preferably something related to topics we discuss in class), study it, and make a presentation to the class as to what the authors did. The presentations should pretty much last an entire class period, but this is subject to negotiation and change.

Grades: The written homework will count 30% of your course grade, the midterm 20%, the final exam 30%, and the presentation 20%.