Syllabus (Content)

Official name of the course:

Elementary Theory of Computation

THEME: In CMSC 351, and other courses, you wanted to solve problems *fast.* This is all well and good! But how to you show that you can't solve it fast? Or that it requires a lot of space? This course looks at models of computation that allow us to show **lower bounds** on how well we can solve a problem.

The following number-of-weeks is approximate. Getting it exact is NP-complete.

1. Regular Languages: DFA's, NFA's, Regular expressions, pumping lemma, Number of states for DFA's and NFA's.

2. Context Free Languages. Chomsky Normal Form. Number of rules in a CFG.

3. P and NP: Turing Machines, Cook-Levin Theorem (SAT is NP-complete). Reductions. Some Complexity Theory. Ways to prove that a problem probably does not have a fast exact solution. Ways around NP-completeness. Classes above NP! Ways to prove that a problem is probably not even in NP!

4. Decidable and enumerable Languages: Turing Machines and the HALTING problem. Ways to show that some problems are undecidable! WS1S is decidable! Mention of Hilbert's 10th problem and Godel's theorem. Might do, Primitive Recurive Hierarchy, Kolg complexity, bounded queries.