

**cs302: Theory of Computation – Spring 2008**  
**Syllabus**

*Computation is what computers do, who needs theory?*

The goal of this course is to understand the fundamental limits on what can be efficiently computed in our universe and other possible universes. These limits reveal deep and mysterious properties about information, knowledge, and processing, as well as practical issues about what can and cannot be computed. We explore these questions by developing abstract models of computing machines and using logic to reason about what they can and cannot compute efficiently.

Two fundamental questions about any problem are:

1. Can it be solved using a given abstract machine? (*computability*)
2. How much time and space are required to solve it? (*complexity*)

In this class we will explore these questions using a series of different computing models.

**Class Meetings.** Tuesday and Thursday, 2-3:15pm in Olsson 120.

**Expected Background.** The official prerequisites for cs302 are cs202 and either cs201 or cs205, all with grades of C- or better.

Students entering cs302 are expected to be comfortable with proof techniques involving first order predicate logic and induction, reasoning about finite and infinite sets, recursive definitions and problem solving, and programming. Students are also expected to have had some previous exposure to asymptotic notation and algorithm analysis (from either cs150 or cs201). If you do not satisfy the prerequisites, you need instructor permission to take the course.

**Textbook.** The required textbook for the course is:

Michael Sipser, *Introduction to the Theory of Computation* (Second Edition).

Two other recommended books that cover much of the same material include:

John Hopcroft, Rajeev Motowani, and Jeffrey Ullman, *Automata Theory, Languages, and Computation*. (Third Edition)

Thomas Sudkamp, *Languages and Machines: An Introduction to the Theory of Computer Science*. (Third Edition)

**Topics.** We will cover most of the Chapters 0-7 of Sipser's book. Topics include: Modeling Computation, automata, regular languages, tag systems, grammars, context-free languages, Turing machines, computability, nondeterminism, complexity classes, NP-Completeness, and reduction proofs. We will also cover some current research topics in theory of computation such as cryptography, on-line algorithms, randomization, and quantum computing. This course is not focused on applications, but we will explore a few applications of the ideas in this class such as bioinformatics, virus detection, Internet searching, social networking, and program analysis.

**Teacher:**

David Evans  
<http://www.cs.virginia.edu/evans>  
[evans@cs.virginia.edu](mailto:evans@cs.virginia.edu)  
Olsson 236A

**Theory Coffee Hours.** I will have “theory coffee hours” Wednesday mornings (starting January 23), 9:30am-10:30am in the Wilsdorf Hall coffee shop. Students in cs302 are welcome to come by to ask any questions you want.

**Office Hours.** I will have “office hours” Monday afternoons (starting January 21), 2:00-3:00pm in Olsson 236A. I am available at other times by email, or you can always stop by when my door is open (which is most of the time when I am here).

**Assistant Coaches:**

Qi Mi (graduate student)  
Suzanne Collier  
Joe Talbott  
Wuttisak Trongsirawat

**Problem-Solving Sessions.** The assistant coaches will hold problem-solving sessions to help students on the problem sets, as well as with other questions about the course material. The schedule and location for these sessions will be posted on the course website.

**Problem Sets.** There will be six problem sets, each containing about 5-10 problems of varying degrees of difficulty. The problem sets are intended primarily for learning, and consistent effort on the problem sets is the most important requirement for succeeding in the class. In most cases, students are encouraged to collaborate in small groups on the problem sets. Each assignment will state a clear collaboration policy, which everyone is expected to follow honorably. The expected topics and due dates for the problem sets is below:

Problem Set 1: Proofs, Finite Automata (Sipser, Chapter 0, Section 1.1) – due  
Tuesday, January 29  
Problem Set 2: Language Hierarchy (Sipser, Chapter 1 (rest), Chapter 2) – due  
Thursday, February 7  
Problem Set 3: Computability (Chapter 3, 4) – due Tuesday, February 19  
Problem Set 4: Time Complexity (Chapter 5, Chapter 7) – due Tuesday, March 18  
Problem Set 5: NP Completeness (Chapter 7.4, 7.5) – due Thursday, March 27  
Problem Set 6: “Hot Topics” – due Tuesday, April 22

**Exams.** There will be two in-class exams and a final.

Exam 1 (covers material through February 19): Thursday, February 28  
Exam 2 (covers material through March 27): Tuesday, April 8  
Final Exam (covers entire course): Saturday, May 3, 9am (scheduled by registrar)

**Extra Credit.** Students are encouraged to go beyond the assigned work. There will be many opportunities to do this including:

“Challenge” Problems – problems will be posed that are too challenging to include in problem sets (in fact, these will often be problems for which no one knows the answer). The first student (or group of students) to offer a satisfactory solution to a challenge problem receives extra credit (typically at least the value of a full problem set). Solvers of challenge problems are also expected to present their solution to the class.

“Communication” Efforts – students will receive extra credit for communicating ideas and stories related to computing theory to the rest of the world. Examples of valuable communication efforts include writing or contributing usefully to an article on a theory of computation topic for Citizendium or Wikipedia; creating a Mathematica demonstration; teaching a short course at an elementary, middle or high school; or creating a web site or some software that illuminates some topic related to theory of computation.

**Grading.** Students will be evaluated based on their understanding of and ability to apply key concepts from the course as demonstrated through the problem sets, exams, contributions in and out class, and extra credit efforts. There is no curve for the course or required grade distribution.

The approximate range of possible weightings is:

<b>Minimum</b>	<b>Default</b>	<b>Maximum</b>	
25%	40%	60%	Problem Sets
10%	15%	20%	Exam 1
15%	15%	30%	Exam 2
20%	30%	50%	Final Exam

Grades will be calculated using the maximum value achieved using several different weightings, including the default weighting.

**Honor.** We assume all students are honorable, and will behave according to the letter and spirit of the honor code. For many assignments in this course, you will be permitted (and encouraged) to work with other students, but there may be rules about what other resources you can use. A high expectation of trust and honor allows us to reuse well known problems, which is to the benefit of everyone involved in the course. The assignments will state the collaboration and resource use policy clearly and you are expected to follow it carefully. If you are unsure if something is permitted, consult with the course instructor before doing it. For exams, students are required to work alone and follow the stated rules exactly. We expect students to do what they can to help their fellow classmates learn, and to help the course staff improve the class.

*If you keep proving stuff that others have done, getting confidence, increasing the complexities of your solutions — for the fun of it — then one day you’ll turn around and discover that nobody actually did that one! And that’s the way to become a computer scientist.*

Richard Feynman, *Lectures on Computation*