Theory of Computation (CS3102) Syllabus
University of Virginia
Professor Gabriel Robins

Course description (as listed in the undergraduate catalog): Introduces computation theory including grammars, finite state machines and Turing machines; and graph theory.

Special emphasis will be placed on basic models, unifying ideas, problem solving, the “scientific method”, as well as elegance, insights, and generalizability in constructing mathematical proofs.

Prerequisites: Discrete mathematics (CS2102) or equivalent

Supplemental reading: How to Solve It, by George Polya, Princeton University Press
Selected papers at: http://www.cs.virginia.edu/~robins/CS_readings.html

Office hours: right after every class lecture, and other times by appointment (also Email Q&A and course blog)

Class structure: two exams (midterm and final), some homework problems taken from the problem sets (which include problems from the textbook and other sources). Extra credit will be given throughout the semester for solving challenging problems.

A brief history of computing:
• Aristotle, Euclid, and Eratosthenes
• Fibonacci, Descartes, Fermat, and Pascal
• Gauss, Euler, and Hamilton
• Boole, De Morgan, Babbage and Ada Agusta
• Venn, Bachmann, Carroll, Cantor and Russell
• Hardy, Ramanujan, and Ramsey
• Godel, Church, and Turing
• von Neumann, Shannon, Kleene and Chomsky

Fundamentals:
• Set theory
• Predicate logic
• Formalisms and notation
• Infinites and countability
• Dovetailing / diagonalization
• Proof techniques
• Problem solving
• Asymptotic growth
• Review of graph theory

Formal languages and machine models:
• The Chomsky hierarchy
• Regular languages / finite automata
• Context-free grammars / pushdown automata
• Unrestricted grammars / Turing machines
• Non-determinism
• Closure operators and non-closures
• Pumping lemmas
• Decidable properties

Computability and undecidability:
• Basic models
• Modifications and extensions to models
• Computational universality
• Decidability
• Recognizability
• Undecidability
• Rice’s theorem

NP-completeness:
• Resource-constrained computation
• Complexity classes
• Intractability
• Boolean satisfiability
• Cook-Levin theorem
• Transformations
• Graph clique problem
• Independent sets
• Hamiltonian cycles
• Colorability problems
• Heuristics

Other topics (as time permits):
• Generalized number systems
• Oracles and relativization
• Zero-knowledge proofs
• Cryptography & mental poker
• The Busy Beaver problem
• Randomness and compressibility
• The Turing test
• AI and the Technological Singularity
Grading scheme:

- **Attendance** 10% (every student is expected to attend all lectures, except for emergencies)
- **Homeworks** 20% (solutions to selected problems will be due several times during the semester)
- **Readings** 20% (various readings will be due each week, as explained below)
- **Midterm** 25% (most midterm questions will be minor variations from problem sets)
- **Final** 25% (most final exam questions will be minor variations from problem sets)
- **Extra credit** 10% (EC given for solving additional problems, and for more readings)

Total: 110% +

Weekly readings: The weekly readings in this class consist of a minimum total of 36 items from the recommended readings list at [http://www.cs.virginia.edu/~robins/CS_readings.html](http://www.cs.virginia.edu/~robins/CS_readings.html) consisting of various papers, videos, animated demos, Web sites, and books. The required ones are highlighted in red font there, while the rest are "electives". The readings item types should constitute a diverse mix, with a minimum of at least 15 videos, at least 15 papers / Web sites, and at least 6 books. Any items above 36 will count towards extra-credit.

The minimum writeup requirements for these readings are a 2 paragraph description for each paper / video / Web site, and 2 page description for books (longer writeups are of course welcomed also). Each writeup should summarize what you learned and what you found interesting / surprising. At least two submissions are due each Monday (by 5pm, beginning the second week of classes), and more than two submissions per week are of course very welcomed and highly recommended (late penalties will be assessed for late submissions). This policy is designed to help you avoid "cramming" at the end of the semester, and also to help you retain more of the knowledge by pacing it more evenly over time. Please Email all readings submissions to the class Email account at homework.cs3102@gmail.com

Study groups: You are encouraged to work on the problem sets and on the homeworks in study groups (of size no more than six people). These study group are intended to foster collaborations, encourage brainstorming, create excitement, and make the learning process more fun. Each study group should meet regularly (say twice per week throughout the semester). Everyone in the study group should contribute fairly to the overall group effort (study groups are not meant for people to just copy solutions verbatim from each other; it’s OK to share ideas and explanations with each other, and then write your own solutions in your own words, but cutting-and-pasting from other people’s work & text is prohibited).

Cheating policy: Cheating and/or plagiarism is strictly prohibited, including under the UVa Honor Code (see [http://honor.virginia.edu/](http://honor.virginia.edu/) for more details). Violators will be subject to serious penalties, including receiving a failing grade on an exam or even in the entire course, as well as possible referral to the UVa Honor Committee (which could lead to expulsion from the University).

Some examples of cheating / plagiarism include:

- Mass-copying of solutions from other people or sources (including downloading from Web sites);
- Mass-sharing of solutions with other people (including uploading to Web sites);
- Cutting-and-pasting from other people’s work and/or text;
- Copying article/book/video/movie reviews from anyone (including from Web sites);
- Having other people solve entire problems for you with little effort on your part;
- Providing other people with verbatim solutions to problems with little effort on their part;

The list above contains only some selected examples of bad behavior and is definitely not exhaustive. It isn’t feasible to enumerate all the possible ways to cheat, so please don’t look for “loop holes” in this policy, and instead please respect its spirit and positive intent. If you’re in doubt as to whether any particular behavior
violates the cheating policy or Honor Code, please ask any of the TAs or the Professor. We would like to encourage honest collaborations, brainstorming, and study groups, and we prefer to not play “gotcha” with well-meaning people. We sincerely ask for your help in creating a more positive learning experience for everyone.

Important note: We do have access to automated cheating/plagiarism detection tools (e.g., software that automatically compares the text of submissions pairwise for textual similarity). These tools also employ automated Google searches that systematically compare phrases and sentences from a submission against general Web searches. We also perform manual comparisons across submissions to detect cheating/plagiarism. So please be aware that cheating/plagiarism is easy to detect, and it is definitely not a risk worth taking.