Instructor: Charles Rackoff  rackoff@cs.toronto.edu
         UTM Office: DH-3088; Tel: 905-828-5352
         Office Hours: M2:00 – 4:00 (usually), or by appointment

Tutor: Jaiganesh Balasundaram


Course requirements: There will be 4 assignments, a midterm, and a Final Exam.
Late homework will not be accepted except for medical or other emergencies.

Students often learn a lot from working with one another, and you are encouraged to meet with other students from class for this purpose. For example, you might work through exercises in the text together or discuss any material you found confusing in lecture or in the readings. It is also legitimate to discuss homework problems with fellow students or to consult other texts. HOWEVER, you must destroy any notes concerning homework problems taken during these meetings and consultations, and you must not write down anything for at least two hours afterwards. Assignments must be written up completely by yourself using only the readings and your own notes as aids. The point is that your written report should be your own work. These rules are meant to ensure that all students understand their solutions to the problems well enough to write up the solutions by themselves. Failure to comply with these guidelines is a serious academic offence.

The Academic Regulations of the University are outlined in the Code of Behaviour on Academic Matters. You are expected to be familiar with, and to abide by, all components of the University's Academic Regulations as defined in the Code of Behaviour on Academic Matters. The document can be found in the UTM Calendar or on the web at https://registrar.utm.utoronto.ca/student/calendar/calendar_detail2.pl?Topic=Discipline Codes

Grading:
Assignments 40%
Midterm 20%
Final Exam 40%
Outline: I hope to cover all or most of the following:

Computability Theory (Chapters 3, 4, 5 in the textbook)
- Turing machines: definitions and examples (section 3.1)
- Other models of computation, the Church-Turing thesis (sections 3.2, 3.3)
- Diagonalization, the Halting problem (sections 4.1, 4.2)
- Decidability and recognizability, examples (sections 4.2, 5.1)
- Reducibility, examples (sections 5.1, 5.2)
- Mapping reducibility, examples (section 5.3)

Complexity Theory (Chapters 7, 8, and parts of 9, 10 in the textbook)
- Models of efficient computation (sections 7.1, 7.2)
- \( P, NP, coNP \), examples (section 7.2, 7.3)
- Polytime reducibility, \( NP \)-completeness (section 7.4)
- Cook's theorem, more \( NP \)-completeness (section 7.5)
- Self-reducibility (not in textbook)
- Space complexity and other complexity classes (sections 8.1, 8.2, 8.3, 9.1, 10.1)