COMP 4601/6601: Models of Computation (Fall 2018)

Time, place: Tuesday/Thursday 11:20am–12:45pm
Dunn Hall 249

Instructor: Thomas Watson
Dunn Hall 315
`Thomas.Watson@memphis.edu`
`http://www.cs.memphis.edu/~twwtson1/`

Office hours: Tuesday/Thursday 1:00pm–2:00pm

TA: Tyler Moore (`tgmoore@memphis.edu`)

Website: `http://elearn.memphis.edu/`

Description: In this course you will learn about how to mathematically define the notion of an algorithm, and how to prove limitations on what algorithms can accomplish and how efficiently they can solve problems.

Topics include: review of proof techniques, finite automata, regular expressions, context-free grammars, pushdown automata, Turing machines, undecidability, reductions, time complexity, NP-completeness.

Prerequisites: COMP 4030 and, by transitivity, COMP 2700

Textbook: Strongly recommended:
Introduction to the Theory of Computation (any edition) by Michael Sipser

Homeworks: There will be eleven homework assignments, each covering the material of about two lectures. See the calendar at the end of this document for the schedule. You may discuss homework problems with other students, but you must write up solutions entirely on your own (and in your own words). You must submit each homework as a single file in the corresponding dropbox folder in the elearn website for the course. If you choose to handwrite your homework solutions (rather than using software such as LaTeX), you may turn in a scan or photo (with all problems combined into a single pdf), as long as the image quality is good enough that the TA will have no problem reading it. If the TA finds it difficult to read one of your solutions, you will get 0 points for that problem.

Homework is due right before the beginning of lecture, and late homeworks cannot be accepted since model solutions will be distributed in class. Each student's lowest two homework scores from the whole semester will be automatically dropped from the final grade calculation, so if extenuating circumstances prevent you from submitting a homework on time, one of these "freebies" will cover it by allowing you to take a 0 without harming your final grade.

Exams: Midterm exam 1 is on October 2nd in class (11:20am–12:45pm, Dunn Hall 249) and will cover homeworks 1–4.

Midterm exam 2 is on November 6th in class (11:20am–12:45pm, Dunn Hall 249) and will cover homeworks 5–8.

The final exam is on December 13th (8:00am–10:00am, Dunn Hall 249) and will be cumulative but with an emphasis on homeworks 9–11.

For each of the midterms you may bring one double-sided sheet of notes, and for the final exam you may bring three double-sided sheets of notes (feel free to use your midterm sheets for two of them). Your sheets of notes may be typed. You may not use anything else during an exam; this means no calculators, textbooks, phones, earbuds, or anything else.

Grading: 5% each for the nine highest homework scores
17% for midterm exam 1
17% for midterm exam 2
21% for the final exam

We will calculate final letter grades in two different ways; then each student will receive the higher of the two letter grades. One way is a fixed grading scale, with the following cutoffs:

A $\geq$ 85%	A– $\geq$ 80%	B+ $\geq$ 75%	B $\geq$ 70%
B– $\geq$ 65%	C+ $\geq$ 60%	C $\geq$ 55%	C– $\geq$ 50%

The other way is a curve, with the following percentages of students receiving each grade:

A: 15%	A–: 15%	B+: 15%	B: 15%
B–: 10%	C+: 10%	C: 10%	C–: 10%

However, we will feel free to give an F to any student who clearly did not put effort into the course (or an A+ to any student with truly exceptional performance).

Cheating: Plagiarism or cheating behavior in any form is unethical and detrimental to proper education and will not be tolerated. All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own work. The plagiarism is incurred when any part of anybody else's work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature on their assignments, but appropriate references must be included for the materials consulted.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructors discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the Office of Student Conduct for further disciplinary action. For further information on U of M code of student conduct and academic discipline procedures, please refer to: `http://www.memphis.edu/studentconduct/misconduct.htm`

Calendar:

Aug 28: lecture 1
Aug 30: lecture 2, hw 1 assigned
Sep 04: lecture 3
Sep 06: lecture 4, hw 1 due, hw 2 assigned
Sep 11: lecture 5
Sep 13: lecture 6, hw 2 due, hw 3 assigned
Sep 18: lecture 7
Sep 20: lecture 8, hw 3 due, hw 4 assigned
Sep 25: lecture 9
Sep 27: lecture 10, hw 4 due, hw 5 assigned
Oct 02: midterm exam 1 (in class)
Oct 04: lecture 11
Oct 09: lecture 12, hw 5 due, hw 6 assigned
Oct 11: lecture 13
Oct 16: Fall break—no class
Oct 18: lecture 14, hw 6 due, hw 7 assigned
Oct 23: lecture 15
Oct 25: lecture 16, hw 7 due, hw 8 assigned
Oct 30: lecture 17
Nov 01: lecture 18, hw 8 due, hw 9 assigned
Nov 06: midterm exam 2 (in class)
Nov 08: lecture 19
Nov 13: lecture 20, hw 9 due, hw 10 assigned
Nov 15: lecture 21
Nov 20: lecture 22, hw 10 due, hw 11 assigned
Nov 22: Thanksgiving break—no class
Nov 27: lecture 23
Nov 29: lecture 24
Dec 04: lecture 25, hw 11 due
Dec 06: study day
Dec 13: final exam (8:00am–10:00am, Dunn Hall 249)

ABET outcomes:

1. Use mathematical notation to specify models of computation.
2. Design finite and pushdown automata.
3. Specify programming language syntax using regular expressions and context-free grammars.
4. Convert between different models of computation.
5. Design Turing machines.

6. Understand the Church-Turing Thesis and the concept of decidability.
7. Use reductions to show that certain problems are undecidable.
8. Analyze the running time of Turing machines.
9. Represent Boolean functions with logic circuits and formulas.
10. Understand the meaning and implications of P vs. NP and NP-completeness.