COMP 4601: Models of Computation (Spring 2022)

Time, place: Tuesday/Thursday 1:00pm–2:25pm
Panhellenic Building (International Center) room 115

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

Office hour: Tuesday 2:30pm–3:30pm

TA: Michael Villarreal (`tmvllrrl@memphis.edu`)

Website: `http://memphis.instructure.com/` (“Canvas”)

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: 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 near 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 assignment folder in the Canvas 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 or zip), as long as the image quality is good enough for the TA to read. 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 scheduled beginning of lecture, and late homeworks cannot be accepted since model solutions will be distributed in class. Timestamps on your computer are not acceptable as proof that homework was completed before the deadline—homework must be submitted on Canvas. Each student’s lowest two homework scores from the whole semester will be 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 letting you take a 0 without harming your final grade.

Exams: The first midterm exam is on February 22nd during class (1:00pm–2:25pm) and will cover homeworks 1–4.

The second midterm exam is on April 5th during class (1:00pm–2:25pm) and will cover homeworks 5–8.

The final exam is on May 5th (10:30am–12:30pm) 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
15% for the first midterm exam
15% for the second midterm exam
20% for the final exam
5% for attendance

Students who attend at least 17 of the 25 lectures (in person) will receive full credit for attendance.

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 $\geq$ 86%	A– $\geq$ 80%	B+ $\geq$ 74%	B $\geq$ 68%
B– $\geq$ 62%	C+ $\geq$ 56%	C $\geq$ 50%	C– $\geq$ 44%

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%

Any student who clearly did not put effort into the course will get an F.

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. Students may not post homework problems to any website or copy solutions from any website (including Chegg and Stack Exchange).

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor's 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.

Calendar:

Jan 18:	lecture
Jan 20:	lecture, hw 1 assigned
Jan 25:	lecture
Jan 27:	lecture, hw 1 due, hw 2 assigned
Feb 01:	lecture
Feb 03:	lecture, hw 2 due, hw 3 assigned
Feb 08:	lecture
Feb 10:	lecture, hw 3 due, hw 4 assigned
Feb 15:	lecture
Feb 17:	review session, hw 4 due
Feb 22:	midterm exam 1 (in class)
Feb 24:	lecture, hw 5 assigned
Mar 01:	lecture
Mar 03:	lecture, hw 5 due, hw 6 assigned
Mar 08:	Spring break—no class
Mar 10:	Spring break—no class
Mar 15:	lecture
Mar 17:	lecture, hw 6 due, hw 7 assigned
Mar 22:	lecture
Mar 24:	lecture, hw 7 due, hw 8 assigned
Mar 29:	lecture
Mar 31:	review session, hw 8 due
Apr 05:	midterm exam 2 (in class)
Apr 07:	lecture, hw 9 assigned
Apr 12:	lecture
Apr 14:	lecture, hw 9 due, hw 10 assigned
Apr 19:	lecture, hw 11 assigned
Apr 21:	lecture, hw 10 due
Apr 26:	review session, hw 11 due
Apr 28:	study day
May 05:	final exam (10:30am–12:30pm)

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.