⟸ COMP 375 Theory of Computation ⟹

M-W-F *Lecture (SC B350)* – 9:30-10:20

Who:	Michael Gousie
Where:	Science Center 1325
When:	Mon 3:30-4:30; Tue 2:30-3:30; Thu 1:30-3:00 and by appointment
E-mail:	`mgousie(at)wheatoncollege(dot)edu`
Web:	`http://cs.wheatonma.edu/mgousie`

Content:

This is a course in (gulp!) theory. But this theory will be presented in the form of many small problems and a larger programming project. This project will let you put theory into practice immediately, thereby connecting what may seem like very abstract ideas into something more concrete.

Theory of Computation deals with *languages*, *grammars*, and *automata* (what the..?). A language is pretty much what you think it is, like C++ or Java. A grammar is a mathematical way of expressing the rules of that language. This is especially useful if you do work or research in programming languages. For instance, a compiler like the one used in Code::Blocks or Dev-C++ reads code and determines if the language's grammar is being followed. We can use an automaton, a theoretical model of a computer, to test ideas/algorithms before writing actual code. Knowing how the model works will show you an elegant way to solve many problems, such as string searching in genomics. Along the way we will do some proofs, which are always good exercise for the mind (yeah, right). Finally, if time permits, we will investigate the solvability question.

Required Text:

Linz, *An Introduction to Formal Languages and Automata*, 5^{th} edition (Jones and Bartlett, 2012).

Recommended Texts:

Rodger and Finley, *JFLAP: An Interactive Formal Languages and Automata Package* (Jones and Bartlett, 2006). This is a workbook/manual for use in conjunction to the JFLAP software we will be using throughout the semester.

Xavier, *Theory of Automata, Formal Languages and Computation* (New Age International, 2004). This is an *e*-book available from Wheaton's library.

Requirements:

There will be a mid-term exam and a comprehensive final exam, together worth 40% of your grade. Five written homework assignments will account for 30% of your grade. Due dates for all homeworks will be announced in class.

Finally, there will be three C++ programming projects. The first two are relatively short warm-up problems worth 5% each. These will be due in the first month of the course. The third project is much longer and more complicated, and will be worth 20% of your grade (yikes!). To make it more manageable, it will be a group project (group size ≤ 3) and it will be broken into several parts that must all work together. The complete project will be due near the end of the semester.

Grading:

Grades will be assigned according to the following scale:

A = 93-100, A− = 90-92, B+ = 87-89, B = 83-86, B− = 80-82, C+ = 77-79, etc.

Exam Schedule:

Exam	Weight	Date
Mid-term exam	20%	March 9
Final	20%	May 11 @ 9 AM

Course Policies:

- You are responsible for all material covered in class, including the reading (shown below).
- If you must miss a quiz or exam for any reason, you must inform me **before** the test. Except in the case of emergency, illness (almost death), or you've found the QA section in the deepest recesses of the library, makeup exams will not be given.
- Homework due dates are **firm**. Homework must be handed in at the start of class on the due date. There are **no** late days for homework.
- The project due dates are **firm** as well. The final code for the projects must be handed in electronically by 11:59:59 PM on the due date, unless noted otherwise on the specification sheet. Any project turned in on the following day will receive a 15% penalty. Anything turned in later will receive a 0. Hard copy and/or written portions must be submitted at the beginning of class the next day or as prescribed in the project handout.
- Programming projects must be written in ANSI C++.[1] Programs will be tested with a standard ANSI C++11 compiler (g11 on Linux).
- You are expected to adhere to the Wheaton Honor Code.
  - Although *discussion* of projects or homework is encouraged, the final, turned-in version should be the result of your own work. This means: **Do not copy any portion of a program!**
  - Collaboration on exams is prohibited.
  - You will be required to write and sign the pledge on all work turned in: *I have abided by the Wheaton honor code in this work.*
  - Any violation of the above guidelines will result in a 0 for that assignment or exam, and/or a failing grade for the course.
- Except when working on an in-class exercise/problem, **the use of a laptop or other computer/pad is not allowed during lecture.** Special arrangements can be made if necessary.
- The use of cell phones, iPods, iPads, iPhones, iPlops, iFlops, and other personal electronic devices is prohibited during class.

[1] Most compilers are either ANSI standard or have a switch to force them to be so. Microsoft compiler products are notoriously bad at complying with standards.

- Accommodations for disabilities:

> *Wheaton is committed to ensuring equitable access to programs and services and to prohibit discrimination in the recruitment, admission, and education of students with disabilities. Individuals with disabilities requiring accommodations or information on accessibility should contact Abigail Cohen, Assistant Dean for Accessibility and Assistive Technology at the Filene Center for Academic Advising and Career Services. ~ cohen_abigail@wheatoncollege.edu or (508) 286-8215 ~*

Course Schedule (subject to change)**:**

Wk #	Week *Begin*	Topic	Reading
	January		
1	21	Introduction, math preliminaries	Chapter 1
2	28	Proofs, DFAs	Section 2.1
	February		
3	4	Finite automata	Sections 2.2–2.4
4	11	Regular languages and grammars	Chapter 3
5	18	Properties of regular languages	Chapter 4
6	25	Context-free languages, parsing	Sections 5.1–5.3
	March		
7	4	CFLs and compilers, mid-term exam (Fri)	Notes
8	11	*SPRING BREAK!!*	Plane tickets to Alta
9	18	Simplification of context-free grammars	Chapter 6
10	25	Pushdown automata	Chapter 7
	April		
11	1	Properties of CFLs	Chapter 8
12	8	Turing machines	Chapter 9
13	15*	More Turing machines	Chapter 10
14	22	Undecidability	Selections from Chapters 11 & 12
15	29	Computational complexity	Selections from Chapter 14
	May		
16	6	Final Exam on May 11 @ 9 AM	Flee!

* No class on April 20th