O COMP 318 Algorithms Θ

M-W Lecture (DC 1315) – 2:00-3:20

Who:	Michael Gousie
Where:	Science Center 1325
When:	Tue 11:00-12:30; Wed 3:30-5:00; Fri 2:00-3:00 (unless meeting) and by appointment
E-mail:	mgousie(at)wheatoncollege(dot)edu
Web:	https://cs.wheatoncollege.edu/mgousie

Content:

Our civilization runs on software.

– Bjarne Stroustrup, lead designer of C++

As future software developers, learning how to write *good* solutions to problems is more important than writing programs that simply "work." If an app doesn't run well, it gets thrown away.

This course is an introduction to the mathematical foundations, design, implementation, and computational analysis of fundamental algorithms. It represents the fourth and final installment of the computer science core. You will polish your C++ skills while learning many of the most-used and important algorithms and techniques available to help solve varied programming tasks. We will cover algorithm analysis in detail so that you can determine which technique is most beneficial in a given situation. This entails both a theoretical understanding of algorithm analysis ("I've proved it will run in $O(n^2)$") as well as practical considerations ("The program took 34 seconds to run"). You will be able to apply these skills in later courses, and indeed, your entire computing career.

Required Text:

- *The Design & Analysis of Algorithms* by Anany Levitin. Addison Wesley; 3^{rd} edition (2012).

Recommended Texts:

- Any text in C++. You may also want to refer back to the Comp 218 text I use: *Data Structures Using C++*, 2^{nd} Edition, by D.S. Malik. Cengage Learning (2010).
- *Introduction to Algorithms*, 4^{th} Edition (or any of the earlier editions), by Cormen, Leiserson, Rivest, and Stein. MIT Press (2022). Rigorous algorithm analysis never goes out of style.
- *Grokking Algorithms*, 2^{nd} Edition, by Aditya Bhargava. Perhaps a more *accessible*, illustrated text.
- *Rant:* While the Interweb, and now AI, are good for looking up specific things, their results are often out of context and not a good source for learning new techniques or understanding the bigger picture.

Grading:

There will be two exams during the semester and a final exam. Together the exams are worth 50% of the grade. Five programming projects account for another 40% of the grade. Written homeworks and possibly in-class work to hand in account for the remaining 10%; note that homework may entail implementing short programs to complete problems.

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
Exam 1	16%	February 25
Exam 2	17%	March 30
Final	17%	May 8 @ 9:00 AM

Project Schedule:

Program	Weight	Topic	Due Date (subject to change)
A1	5%	C++ warm up	February 5
A2	8%	Using large data files; analysis	March 1
A3	9%	Algorithm implementation	March 26
A4	9%	Trees	April 14
A6	9%	TBD	April 30

Course Policies:

- You are responsible for all material covered in class.
- You are responsible for completing all of the reading, noted 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, or you've gotten irretrievably lost using the library entrance in the Discovery Center, makeup exams will not be given.
- Programming projects will be completed in C++. You may work on your programming assignments on any platform and any compiler. However, the final turned-in version must work properly using an ANSI standard C++17 compiler. Be especially careful if you use a Microsoft compiler (Visual Studio), as these are often non-standard.
- Written homeworks should be neat and done on loose-leaf or plain paper. Do not tear pages out of a notebook that yield "fringes" on the paper. Staple multiple pages together.
- Assignment due dates are **firm**.
  - All programming projects must be submitted in Canvas by 11:59:59 PM on the due date unless noted otherwise. Projects submitted on the following day will receive a 15% penalty. Anything turned in later will receive a 0. Any required hard copy and/or written portions must be submitted at the **beginning** of class on the next day or as instructed on the specification sheet.

  - Projects may be turned in early! You can also resubmit your project before the deadline if you find an error in an earlier submission. Only the last submitted project will be graded.
  - All written homework must be submitted at the **beginning** of class on the due date, or as specified on the assignment sheet. There are **no** provisions for late homework.
  - A computer crash is not an excuse for late work. It is important that you **back up all of your work!**
  - There will not be any individual "extra credit" work. If you did not have time to do a good job on the original assignment, how will you have time to do *additional* work?

- You are expected to adhere to the Honor Code. (See https://wheatoncollege.edu/about-wheaton-college/honor-code/)
  - Although *discussion* of assignments is encouraged, the *implementation* of programs or the working out of problems is to be the result of your own work. Any copying of programs or portions of programs will engender penalties.
  - AI can help you speed up the programming process by having it do some of the more menial tasks. It can also solve homework problems for you. However, your program or homework should still be **your own work**. Copy/paste is not the way to learn how to program, whether you are doing this from another person or from an AI application.

    If you are unsure where the line is between collaborating with AI and copying from AI, we recommend the following heuristics:
    * Never hit *Copy* within your conversation with an AI assistant (and then *Paste* into your code). You can copy your own work into your conversation, but do not copy anything from the conversation back into your assignment. Instead, use your interaction with the AI assistant as a learning experience, then let your assignment reflect your improved understanding.
    * Do not have your assignment and the AI agent itself open on your device at the same time. Similar to above, use your conversation with the AI as a learning experience, then close the interaction down, open your assignment, and let your assignment reflect your revised knowledge. This heuristic includes avoiding using AI assistants that are directly integrated into your composition environment: just as you should not let a classmate write content or code directly into your submission, so also you should avoid using tools that directly add content to your submission.
  - If a program or homework looks suspicious, I may ask you to explain the purpose, function, and details of your code or answer; if you can't, it will be considered plagiarized.
  - Written homework/papers should absolutely be your own work.
  - Collaboration on exams is prohibited.
  - Any violation of the above guidelines will result in a 0 for that assignment or exam, and/or a failing grade for the course.
  - 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.

- Except when an in-class exercise/problem is being worked on, **the use of a laptop or other computer/tablet which requires typing is not allowed during lecture.** It is to your advantage to take

notes and draw diagrams using pen and paper, as this has been shown to be more effective than typing. Special arrangements can be made if necessary (come see me).

- The use of cell phones, iPhones, iPods, iPads, iPlops, iFlops, and other personal electronic devices is prohibited during class, labs, and exams.
- Please do not leave once class has begun. This is distracting to the class and to the instructor.
- Accommodations for disabilities:

> *Wheaton College is committed to providing equitable access and supportive services for all students to fully access and thrive in the academic, residential and social aspects of student life. Affirmatively, Wheaton provides appropriate accommodations for eligible students with documented disabilities to afford equal access to educational programs and services. Individuals with disabilities and other access concerns requiring accommodations or information on accessibility should reach out to Accessibility Services at the Filene Center:*
> *∼ accessibility@wheatoncollege.edu or (508) 286-3794 ∼*

Course Schedule **(Subject to change)**

Week	Day	Topic	Reading
Week 1		**Introduction**	Chapter 1
	Jan 21	Introduction Remember C++??	
Week 2		**C++ features-n-fun**	C++ text
	Jan 26 Jan 28	Advanced C++ concepts Mathematical notation and theorems	 handout
Week 3		**Analysis of algorithms**	Chapter 2
	Feb 2 Feb 4	Formal algorithm analysis O, Ω, and Θ notation (Um, huh?)	
Week 4		**More analysis of algorithms**	Chapter 2, cont.
	Feb 9 Feb 11	Non-recursive and recursive analysis Algorithm visualization and sorting	
Week 5		**Brute force algorithms**	Chapter 3
	Feb 16 ⇒Feb 18⇐	Sorting **No class – conference**	
Week 6		**Exhaustive search**	Chapter 3, cont.
	Feb 23 Feb 25	Convex Hull #1 (uh, what?), traveling salesman Exam 1	

Course Schedule, cont.

Week	Day	Topic	Reading
Week 7		**Decrease-and-conquer**	Chapter 4
	Mar 2 Mar 4	DFS, BFS Topological sort, computing median	
Week 8		**SPRING BREAK**	
	⇒Mar 9-13⇐	**No class**	
Week 9		**Divide-n-conquer**	Chapter 5
	Mar 16 ⇒Mar 18⇐	Mergesort/Quicksort **No class – MAP Day**	
Week 10		**Divide-n-conquer**	Chapter 5, cont.
	Mar 23 Mar 25	Binary trees, fast integer math Convex hull #2 (AGAIN??)	
Week 11		**Transform-n-conquer**	Chapter 6
	Mar 30 Apr 1	Exam 2 AVL and Red-black trees	 notes
Week 12		**More transform-n-conquer**	Chapter 6, cont.
	Apr 6 Apr 8	2-3 trees, B-trees Heaps, Horner's Rule	Section 7.4
Week 13		**Space and time**	Chapter 7
	Apr 13 Apr 15	String matching, Harspool's Hashing	Section 3.2
Week 14		**Dynamic programming**	Chapter 8
	Apr 20 Apr 22	Warshall's, Floyd's Dijkstra's, Prim's	
Week 15		**Greedy algorithms**	Chapter 9
	Apr 27 Apr 29	Kruskal's, Huffman code P, NP, NP-Complete (Uh, what now...?)	 Section 11.3
Week 16		**Final Exam Week**	
	May 8	Final exam @ 9:00 AM	