
★ COMP 218	Data Structures	★
<i>M-W-F</i>	<i>Lecture (DC 1313) – 10:30-11:20</i>	
<i>M</i>	<i>Lab (DC 1313) – 3:30-5:20</i>	

Who: Michael Gousie
 Nick Almeder (lab section)
 Where: Discovery Center 1325
 When: Tue 1:30-3:00; Wed 3:30-4:30; Fri 11:30-12:30
 and by appointment
 E-mail: [mgousie\(at\)wheatoncollege\(dot\)edu](mailto:mgousie@wheatoncollege.edu)
 Web: <http://cs.wheatoncollege.edu/mgousie>

Content:

This course is next in the core computer science sequence, following COMP 118. The main objective is to delve into more serious study of computer science (CS). This will involve:

- Continuing with C++, a powerful, general-purpose programming language. It is one of the most popular languages and is used by many of the top companies. After gaining experience in C++, it is generally easy to learn to another language, such as Java.
- Learning more Object-Oriented Programming (OOP) paradigms, and the various C++ features that make such programming possible.
- Honing your problem-solving skills. As we cover more C++ and OOP, the problems get harder as your ability to solve them grows.
- Adding various data structures to your toolbox. These techniques will further enhance your ability to solve more complex problems.
- Getting a taste of theoretical computer science, in the form of “Big O” efficiency analysis. This introduction will help you move on to Algorithms, the culmination of the CS core.

Required Text:

Malik, D.S. *Data Structures Using C++, 2nd Edition*. Cengage Learning, 2010.

Recommended Texts:

Stroustrup, Bjarne. *The C++ Programming Language, 4th Edition*. Addison-Wesley, 2013.

Lischner, Ray. *Exploring C++: The Programmer’s Introduction to C++*. Apress, 2009. Available on the course web site.

Various. *C++ Programming*. Wikibooks.org, 2012. Available on the course web site.

Requirements:

I expect a lot from my students. Why? Because I like to challenge you to learn as much as possible and expand your boundaries. And because *you can do it!*

There will be two exams during the semester and a comprehensive final exam. The final exam is worth a higher percentage of your grade because the extra time during the final will allow for more in-depth questions that combine many of the concepts that have been covered during the

semester. The exams make up 50% of your grade. There will be 7 programming assignments as well, following the schedule shown below. The projects are worth 46% of your grade. The remaining 4% of your grade will be comprised of lab quizzes which entail **completing** a small program during the allotted lab period or answering a few, short questions. There may also be some short written homework problems.

Exam Schedule:

Exam	Weight	Date
Exam 1	15%	September 27
Exam 2	15%	November 4 (during lab period)
Final Exam	20%	December 9 @ 9:00 AM

Assignment Schedule:

Program	Weight	Topic	Due Date (subject to change)
DS1	4%	Operator overloading; I/O	September 12
DS2	7%	Inheritance and more	September 24
DS3	8%	Linked lists	October 10
DS4	7%	Recursion	October 27
DS5	7%	Stacks	November 10
DS6	7%	Queues	November 26
DS7	6%	Trees/Hashing	December 5

Grades:

Grades will be assigned according to the following scale:

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

Course Policies:

- You are responsible for all material covered in class, including the reading shown in the schedule below.
- Regular attendance to the afternoon lab period is required. There will be no makeup lab quizzes.
- 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 got lost in the original swimming pool,¹ 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 such as Visual Studio.
- We will use some AI to show its strengths and weaknesses in the pursuit of a well-written, efficient program. See also the Honor Code section about using AI, below.

¹Do you know where this is?

- Assignment due dates are **firm**.
 - All programming projects must be submitted electronically by 11:59:59 PM on the due date unless otherwise noted. Projects submitted on the following day (i.e., after midnight) 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.
 - Written homework must be handed in at the start of class on the due date. There are **no** provisions for late homework.
 - 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.
 - 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?
- A computer crash or inadvertently losing your files is not an excuse for late work. It is important that you **back up all of your work!** A flash drive is useful for this.
- You are expected to adhere to the Honor Code. (See <https://wheatoncollege.edu/about-wheaton-college/honor-code/>)
 - Although *discussion* of projects, homework, and lab work is encouraged, the *implementation* of programs is to be the result of your own work. This means: **Do not copy any portion of a program!**
 - AI can help you speed up the programming process by having it do some of the more menial tasks. However, your program 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 looks suspicious, I may ask you to explain the purpose, function, and details of your code; if you can’t, it will be considered plagiarized.
 - Collaboration on exams is prohibited.
 - Any violation of the above guidelines will result in a 0 for the assignment/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*. Instructions for electronic submissions will be given in the project specifications.
- **The use of a laptop or other computer/pad is not allowed during lecture** unless we are working on code as a group. Special arrangements for electronic note-taking can be made if necessary.
- The use of cell phones, iPods, iPads, iPhones, iPlops, iFlops, and other personal electronic devices is prohibited during class, lab, and exams. I encourage you to turn off your device and put it away during class and whenever you are programming. The distraction of these devices is just harming your own learning experience.
- Please plan your restroom breaks so that you will not disrupt class; this means that you should not leave class once you arrive.
- 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):

Wk #	Week Begin	Topic	Reading*	Lab
	August			
1	25	Intro, C++ review, ADTs	Chapter 1, C++ reference	No lab
	September			
2	1	Object oriented design (OOD): operator overloading & inheritance	Chapter 2, online examples	C++ review
3	8	Pointers and more	Chapter 3	Op. overloading
4	15	Linked lists, templates	Chapter 5	Pointer fun
5	22	Standard Template Library (STL)	Chapter 4	Linked lists
6	29	Exam 1 RECURSION	Chapter 6	Templates/STL
	October			
7	6	More recursion	Notes	Recursion
8	13	----- <i>No class Oct. 14 – October Break</i> Stacks <i>MAP Day: Oct. 17</i> -----	Chapter 7	No lab
9	20	Queues	Chapter 8	Stacks
10	27	Polymorphism: virtual functions	Notes	Queues
	November			
11	3	Searching and “Big O” analysis	Chapter 9	Exam 2
12	10	Hashing <i>Last day to withdraw: Nov. 15</i>	Chapter 9, cont.	“Big O”
13	17	Sorting and “Big O” analysis	Chapter 10	Hash functions
14	24	Trees <i>No class Nov. 27/29 – Thanksgiving</i>	Chapter 11	BST
	December			
15	1	More trees, review	Chapter 12	TBD
16	8	Final Exam: Monday, Dec. 9 @ 9:00 AM		

*Readings with chapters come from *Data Structures Using C++* by Malik.