{ COMP 118 Object-Oriented Programming }

M-W-F	*Lecture (DC 1315)* – 11:30-12:20
W	*Lab (DC 1315)* – 3:30-5:20

Who:	Michael Gousie
Where:	Discovery Center 1325
When:	Mon 2:30-4:00, Tue 11:00-12:30; Fri 2:00-3:00 (unless meeting) and by appointment
E-mail:	mgousie(at)wheatoncollege(dot)edu
Web:	http://cs.wheatoncollege.edu/mgousie

Content:

In this course, you will learn to program in **standard** C++, with an emphasis on its object-oriented features. This will prepare you for the next course in the major sequence, Data Structures, as well as most other upper-level courses. Lab periods will entail writing small programs that cover the week's topic.

In particular, we will cover:

- Basic types, I/O, and sequential statements
- Control statements
- Functions, parameter passing, and scope
- Dynamic memory allocation
- Arrays, vectors, and other basic data structures
- Principles of object-oriented programming (OOP)
- How to implement object-oriented programs in C++
- Function and operator overloading
- Recursion, templates, inheritance (overview)

Recommended Texts:

Downey. *Think C++* (Published online, 2022). This is available through a link on the course web page.

Goldwasser and Letscher. *A Transition Guide from Python 2.x to C++* (Published online, revised 2011). Available on the course web page.

Lischner. *Exploring C++: The Programmer's Introduction to C++* (Apress, 2009). Available on the course web page.

Requirements:

There will be two exams during the semester and a comprehensive final exam. The exams are worth 45% of your grade. Exams will be **written**; a computer (or any other electronics) will **not** be used for any exams. The goal is to *learn how to program*, not to learn how best to search the web for an answer. There will be several lab quizzes as well, worth 5% of your grade. In order to learn to program, one must practice! Therefore, there are six programming projects (ack!), which comprise 50% of your grade.

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	Time
Exam 1	10%	February 23	In class
Exam 2	15%	April 3	During lab period
Final	20%	Friday, May 10	9:00 AM

Programming Assignments:

Program	Weight	Topic (Subject to change)	Due Date
P1	5%	Basic C++	February 8
P2	7%	Control flow	February 20
P3	8%	Functions	March 7
P4	10%	Arrays	March 31
P5	10%	Classes	April 14
P6	10%	Operator overloading	May 2

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, or you found aliens in Wheaton's original pool (Where is that?), makeup exams will not be given.
- Programs will be written in C++ using the CLion IDE. Other IDEs are acceptable, but please see me before using them. Visual Studio is especially problematic, and is **not recommended**.
- 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 will receive a 15% penalty. Anything turned in later will receive a 0. Hard copy must be submitted the following day or as indicated in the program specifications.

  - Although deadlines are firm, please contact me **beforehand** if there are extraordinary circumstances.
  - Projects may be turned in early! You can also resubmit projects any number of times before the deadline if you find an error in an earlier submission.
  - 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.
  - Although *discussion* of projects or homework is encouraged, the final *implementation* of programs should be the result of your own work. Any copying of programs or homework is prohibited.
  - 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 the assignment/exam and/or a failing grade for the course.

- You should take *written* notes during class. Research has shown that written notes are more effective in learning than typed notes. Of course, we will do some programming in class, so in those circumstances, code will be typed on your computer. Otherwise, typing should not be done during lecture.

- The use of cell phones, iPods, iPads, iPhones, iPlops, iFlops, and other personal electronic devices is prohibited during class, lab, and exams.

- 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
	January			
1	21	Introduction, problem solving	*Guide*: Ch. 2, *Think*: Ch. 1	CLion
2	28	Basic C++, types, I/O	*Guide*: Chs. 3, 4, 6 *Think*: Ch. 2	I/O
	February			
3	4	Flow of control	*Guide*: Ch. 5 *Think*: Chs. 4.1-4.5, 6	Control
4	11	File I/O, functions	*Guide*: Ch. 6 *Think*: Ch. 5, 15.1-15.5	Files
5	18	Exam 1, pointers	*Guide*: Ch. 8	Functions
6	25	Scope, arrays	*Guide*: Ch. 8 (cont.)	Arrays
	March			
7	3	Memory management, n-D arrays	*Guide*: Ch. 8 (cont.)	Dynamic arrays
8	10	*SPRING BREAK*		Powder skiing
9	17	Structs, intro to OOP	*Guide*: Ch. 7	
	⇒3/20⇐	**No class! MAP Day!**		TBD
10	24	Classes, operator overloading	*Think*: Ch. 11**	Classes
11	31	More overloading, header files	*Guide*: Ch. 12	Exam 2 (Weds)
	April			
12	7	Recursion	*Think*: Ch. 5	More objects
13	14	Vectors, templates	*Guide*: Chs. 9, 10 *Think*: Ch. 10	Recursion
14	21	Inheritance	Notes	Vectors
15	28	Review		OOP
	May			
16	5	Final Exam, May 10 @ 9 AM	Happy Summer!	

* *Guide* refers to *A Transition Guide from Python 2.x to C++*; *Think* refers to *Think C++*

** Chapter 11 requires understanding structures from Chapters 8 and 9.