CSC 319: Object-Oriented Design and Programming

3 credits, Fall 2017, *revised August 25, 2017*

Class Times:	AuSable 113	Mon., Wed. and Fri. 11:00 a.m.–11:50 a.m.
Final exam:	To be scheduled	
Instructor:	Dr. Steven P. Crain	
Email:	`Steven.Crain@plattsburgh.edu`	
Office Hours:	AuSable 310	Mon. 4:30–5:30 p.m.
	Phone 518-564-2782	Wed.10:00–10:50 a.m.
		Thu. 2:30–3:30 p.m.
		Fri. 10:00–10:50 a.m.

Course Description

Fasten your seatbelts (design principles?) and prepare for lift off: this course is where your programming skills should leave the ground! You will learn to apply an object-oriented design process to create complex applications without getting lost in the details. This course requires substantial investment of time programming as you learn C++ and advanced programming techniques. Your work will be supported by a UML design tool, automatic code generation, an integrated development environment, revision control, debugging tools and profiling tools.

The main topics of the course are:

Programming methodologies
Object-oriented design
Design principles
Desgin patterns
UML diagrams
C++ program structure and syntax
Primitive datatypes
Standard libraries
Templates
GUI programming
Modern IDE
Debugging
Memory management
Managing large programs
Git
Exception handling
Classes
Class Hierarchies
Function and operator overloading
References and Pointers
Event handling
Secure coding techniques
UML tool with code generation
Profiling

Students are expected to have familiarity with C, data structures and algorithms. CSC223 and CSC309 provide the typical preparation. Students should not attempt this course without this foundation.

Learning Objectives

1. Apply the object-oriented paradigm to design a complex program.

2. Develop ability to write complex programs in C++.

3. Learn to use advanced software modeling and development tools.

Textbooks

- **Required:** Stroustrup, B. *The C++ Programming Language, 4th ed.*, Addison-Wesley, 2013. Older editions are not adequate.

Course Requirements

It is impossible to learn object-oriented design and programming without substantial practice. As such, plan to expend more effort in this class than is necessary in most other courses, about 6 hours per week outside of class.

Reading

You should read the relevant portions of the textbook both before and after each lecture. This will take about an hour per week for most students.

Class Attendance and Participation, 15%

This grading component is divided roughly into half attendance and half participation in discussions and lab activities. Being late to class twice is counted the same as missing class once.

Labs, 55%

There will be up to one lab assignment due each week. The labs will vary in difficulty, and some will require substantial effort. Most of the early labs must be done individually. Later labs will involve group work.

Please be aware that I often find evidence that students worked together when they shouldn't have. If you are caught cheating, the Academic Dishonesty Policy provides a strict procedure that may result in penalties up to expulsion.

Midterm Exam, 15%

This class will have a midterm exam on 10/6, the Friday before fall break. Plan accordingly. The exam will test your knowledge of subject matter, including writing correct C++ code without computer support.

Final Exam, 15%

This class will have a final exam. The final exam will not be cumulative.

Students with Disabilities, English as a Second Language Students

If you have a disabling condition or English is not your native language, you must register with Student Support Services — Angell College Center, 564-2810 prior to requesting accommodations.

Plagiarism and Academic Dishonesty

I have very low tolerance for academic dishonesty, and will vigorously pursue available remedies for any incidents. It is expected that all students enrolled in this class support the letter and the spirit of the Academic Honesty Policy as stated in the College Catalog. Specifically I expect that:

- Exams will be closed book, closed notes, and no communication between students. This includes discussing the exam with students who are taking the exam at another time.
- Discussion of assignments is expected and encouraged, however all work and code on assignments should be your own without outside assistance.
- Sources should be cited including the textbook and other web sites when you use them in your work.
- You are not permitted to share your source code with other students, including future ones.
- You are not permitted to use other students' solutions as your own (even those from a prior semester), nor answer keys, nor instructor versions.

Illustrative examples:

- Confirming that we had an exam is OK, but telling another student in the class who has not taken it what topics were covered or even whether it was easy or hard is **NOT OK**.
- On a project or homework, discussing what needs to be done and how it can be done is OK. Having a student (other than a TA) go over your code is **NOT OK**. Discussing what might be wrong and how to tell is OK (and encouraged).
- Unless otherwise specified in the assignment, using insertion sort code based on the version from your Java textbook is OK as long as you give appropriate credit, /* based on insertion sort from Weiss 3rd Ed, p. 306 */.

Note: Special thanks to Ben Kuperman, from whose syllabus parts of this syllabus were derived.

Schedule

The following is the approximate schedule for the material in this course.

Date	**Reading**	**Topic**
8/28		Introduction to the course
8/30	Sec. 2.2, 2.4	Basic C++
9/1	Sec. 2.3	Basic C++ classes
9/4	Sec. 4.1–3	Standard C++ libraries, strings, stream I/O
9/6	Sec. 4.4	Standard C++ containers
9/8	Sec. 4.5, 11.4	Container algorithms
9/11	Sec. 6.1–3	C++ datatypes
9/13	Sec. 6.4–5	Object lifecycles
9/15	C. 7	Pointers and references
9/18	Sec. 11.2, 34.3	Memory allocation
9/20		Valgrind
9/22	Sec. 16.2	Classes
9/25	Sec. 16.3	Concrete classes
9/27	Sec. 20.2	Derived classes
9/29	Sec. 20.3-4	Abstract classes and virtual functions
10/2		Debugging strategies
10/4		Review session
10/6		Midterm Exam
10/9		**NO CLASS: Fall break**
10/11		GDB
10/13	Sec. 19.4, 20.5	Access control
10/16–20	Sec. 21.2	Inheritance Patterns (Implementation, interface, adapter, template, inversion of contr
10/23–11/3		UML design and ArgoUML
11/6–9		wxWidgets
11/13–17		Object-oriented programming
11/20		Unit tests
11/22–24		**NO CLASS: Thanksgiving**
11/27		Advanced Git
11/29		Profiling
12/1	Ch. 23	Templates
12/4–6	Ch. 24-25	Generic programming
12/8		Review
Final exam period		Final exam