CSCI 140.02 (42359) Syllabus
C++ Language & Objects
Spring 2022

Lecture

Time: Mo/We 11:10 am - 12:35 pm
Room: 61-3311

Lab

Time: Mo/We 12:45 pm - 2:10 pm
Room: 61-3311

Instructor

Professor Dominick Atanasio (datanasio@mtsac.edu)
Office: 61-1656
Hours: Mo/We 3-4pm | Tu/Th 9:30-10:30am
Or by appointment via Zoom

Prerequisites

CSCI 110 or equivalent C/C++ programming.

Textbook, References

Textbook:

- **Deitel, Harvey M., and Paul J. Deitel. C++ How to Program. Pearson, 2017. (9780134448237)**

References (not required):

- Lippman, Lajoie, and Moo. C++ Primer 5E. Addison-Wesley Professional. (9780321714114)

Course Description

This course is for computer science, mathematics, engineering, and other science students. You will learn C++ programming and object-oriented paradigm, control structures, functions, arrays, pointers and strings, classes and data abstraction, C++ object programming, operator overloading, inheritance, virtual functions and polymorphism, stream input and output, templates, exception handling, file processing, data structures in C++, string processing and recursion.

Course Objectives:

Upon the completion of this course, the student shall acquire the following knowledge and skills:

- Students will be able to analyze problems and design algorithms in pseudocode.
- Students will be able to read, understand and trace the execution of programs written in C++ language.
- Students will be able to use given classes and virtual functions in a class hierarchy to create new derived classes and the code that uses them.
- For a given algorithm students will be able to write modular C++ code using classes in an OOP approach.

Measurable Outcomes

- Analyze problems and design appropriate algorithms.
- Code algorithms into the C++ language.
- Recognize and produce proper C++ syntax.
- Use correct data type and data structures, including objects, linked lists, stacks, and queues.
- Utilize recursion, iteration, arrays, pointer arithmetic.
- Demonstrate the paradigm of object-oriented programming.
- Write, organize, and assemble program documentation.
- Utilize encapsulation, overloading, inheritance, and polymorphism.
- Utilize data abstraction, separate program interface and implementation
- Develop standards for comparing the efficiency of various algorithms.
- Demonstrate debugging techniques.

Required Resources

- A reliable Internet connection to access Canvas and email.
- Text Editor or C++ IDE (Visual Studio Code)
- G++ Compiler
- Note-taking materials

Grading

Midterm Exams (3)	30%
Final Exam (Mandatory)	25%
Lab Final Exam (Mandatory)	20%
Labs (8) and Projects (3)	25%

Letter Grades			
A	[90, 100]	D	[60, 70)
B	[80, 90)	F	[0, 60)
C	[70, 80)		

Lab Activities

The goal of the lab is to enhance teaching/learning and to develop a deeper understanding of concepts taught in class. We will use lab time to lecture on topics, demonstrate new concepts via programming examples, and you would have time to try out new concepts under the supervision of your professor. Lab activities would include going over new concepts, solving problems, modifying existing programs, testing programs, debugging programs, and conducting experiments. Important concepts will be demonstrated so you can complete your programming assignments. Besides coding analysis, design, testing and debugging it will be important that you learn individually or in collaborative group work. You might be asked to present/share your solutions with other students so giving you a chance to learn from others.

Students need to prepare for lab by reading the material assigned and covered in lecture. Each lab will start with a preview of the concepts on which the lab is based in one of the following forms: instructor lecture or demonstration, class questions or discussion, using class worksheets: T/F, fill-in blanks, or short answers. Class lab handouts will guide you to use the lab resources, lab textbook or interactive visualization software. Handouts on specific additional topics, code skeletons or demonstrations will be provided.

As you work on code, it is recommended that you talk to each other (unless otherwise instructed) or ask your instructor for individual help. If a topic will be of interest for the whole class, we will take it up for class discussion and your instructor will lecture or demonstrate to clarify that topic. When you create your code, your work will be supervised, and your instructor will take questions to make sure you understand the concepts. Programming will be done on mostly MS Windows OS, but we might explore another OS as well. Your answers to questions on the lab handouts will provide feedback and help accessing your work. You will be asked to formulate your own conclusions in lab worksheets, and we shall use discussions, presentations, or lectures to present/debate these conclusions.

Student Responsibilities

Attendance Policy

Attendance is mandatory. Missing more than four lectures or labs will be cause for being dropped from the class. **Students are responsible for any material covered in class.** Some of the material covered in class may not be in the required textbook or in the lecture notes yet given during the lecture; you will still be responsible for this information.

Lectures

All lectures will be given synchronously and in person. If you miss a lecture you may have to ask a classmate for notes, this is your responsibility. Lectures may include coding examples. You are encouraged to code along with the instructor and to ask questions.

Exams

Exams will be given on the scheduled date. You are required to attend class on the day of the exam. When taking the exam, you must remove from your desk all items that are not part of the exam process. You must also remove from your person any smart devices like phones, watches, personal computers, audio devices, etc. Phones should be muted during the exam.

Late Submission and Make-up Exams

There will be no late submissions accepted; ample time will be provided for each assignment. Make-up exams for absences **may be allowed only at the discretion of the instructor**. You are required to inform the instructor well before the exam if you are going to miss it. A valid reason, and possibly some form of proof, will be required. There will be no allowance for missing the final exam.

Announcements and Emails

Announcements will be made in class, via email, and/or on Canvas. It is your responsibility to check Canvas and your Mt. SAC email frequently throughout the term. **Assignments may be posted in class or to Canvas without email notification**.

Academic Misconduct

During exams, no discussion of any kind (except with the instructor) is allowed. No access to any type of material (electronic or otherwise) or use of devices, unless provided by the instructor, is permitted. For assignments, I encourage discussion among students, but I expect each student to hand in their own original work.

You are responsible for doing your own work and for ensuring that your work is not copied by others. Share your ideas, not your work. <u>No excuses for academic dishonesty will be accepted</u>. Please read our Student Life website concerning Academic Honesty (https://bit.ly/3FLNJIm) or read pages 440-441 of the 2021-22 Mt SAC catalog.

Please refer to our Canvas site for all other information concerning this class.

Tentative Course Schedule

Week	Date	Topic
1	02/21	**Holiday: Presidents Day**
	02/23	Topic 1: Introduction to Computers and C++
2	02/28	Topic 1: Introduction to Computers and C++
	03/02	Topic 1: Introduction to Computers and C++
3	03/07	Topic 1: Introduction to Computers and C++
	03/09	Topic 1 & Topic 2
4	03/14	Topic 2: Algorithm Development and Flow Control
	03/16	Topic 2: Algorithm Development and Flow Control
5	03/21	Topic 2: Algorithm Development and Flow Control
	03/23	Topic 2: Algorithm Development and Flow Control
6	03/28	**Exam 1**
	03/30	Topic 3: Functions and an Intro to Recursion
7	04/04	Topic 3: Functions and an Intro to Recursion
	04/06	Topic 3: Functions and an Intro to Recursion
8	04/11	Topic 4: Arrays and Pointers
	04/13	Topic 4: Arrays and Pointers
9	04/18	Topic 4: Arrays and Pointers
	04/20	Topic 5: Classes
10	04/25	**Exam 2**
	04/27	Topic 5: Classes
11	05/02	Topic 5: Classes
	05/04	Topic 6: Object-Oriented Programming
12	05/09	Topic 6: Object-Oriented Programming
	05/11	Topic 6: Object-Oriented Programming
13	05/16	**Exam 3**
	05/18	Topic 7: File Processing
14	05/23	Topic 7: File Processing
	05/25	Topic 8: Exception Handling
15	05/30	**Holiday: Memorial Day**
	06/01	Topic 8: Exception Handling
Final Exam	**06/06**	**Monday 10:30am - 1:00pm**
Final Lab Exam	**06/08**	**Wednesday 10:30am - 1:00pm**

Week	**Date**	**Topic**
1	02/21	**Holiday: Presidents Day**
	02/23	Topic 1: Introduction to Computers and C++
2	02/28	Topic 1: Introduction to Computers and C++
	03/02	Topic 1: Introduction to Computers and C++
3	03/07	Topic 1: Introduction to Computers and C++
	03/09	Topic 1 & Topic 2
4	03/14	Topic 2: Algorithm Development and Flow Control

	03/16	Topic 2: Algorithm Development and Flow Control
5	03/21	Topic 2: Algorithm Development and Flow Control
	03/23	Topic 2: Algorithm Development and Flow Control
6	03/28	**Exam 1**
	03/30	Topic 3: Functions and an Intro to Recursion
7	04/04	Topic 3: Functions and an Intro to Recursion
	04/06	Topic 3: Functions and an Intro to Recursion
8	04/11	Topic 4: Arrays and Pointers
	04/13	Topic 4: Arrays and Pointers
9	04/18	Topic 4: Arrays and Pointers
	04/20	Topic 5: Classes
10	04/25	**Exam 2**
	04/27	Topic 5: Classes
11	05/02	Topic 5: Classes
	05/04	Topic 6: Object-Oriented Programming
12	05/09	Topic 6: Object-Oriented Programming
	05/11	Topic 6: Object-Oriented Programming
13	05/16	**Exam 3**
	05/18	Topic 7: File Processing
14	05/23	Topic 7: File Processing
	05/25	Topic 8: Exception Handling
15	05/30	**Holiday: Memorial Day**
	06/01	Topic 8: Exception Handling
Final Exam	**06/06**	**Monday 10:30am - 1:00pm**
Final Lab Exam	**06/08**	**Wednesday 10:30am - 1:00pm**