CSC112: Introduction to Computer Science II

Dr. Robert Lowe

Spring, 2020

E-mail: robert.lowe@maryvillecollege.edu Office Phone: 865-981-8169
Office Hours: MWF 1:00PM – 2:00PM, TR 3:00PM – 4:00PM Class Hours: MW 2:00 – 4:00
Office: SSC 214 Class Room: SSC 204

Course Description

A continuation of Computer Science 111 with emphasis on advanced programming features. Laboratory work supplements and expands lecture topics and offers supervised practice using programming.

Required Materials

- *Big C++. 3/e.* Cay Horstmann. https://tinyurl.com/bigcpp
- An internet connected computer of some sort.

Prerequisites

CSC111

Course Goals

- Gain an advanced understanding of Object Oriented Programming.
- Learn to use Object Oriented Analysis and Design to build large complex programs.
- Gain a preliminary understanding of low level programming concepts, especially memory addressing.
- Increase your knowledge of using and programming the UNIX operating system.
- Hone your knowledge of programming tools like make, g++, and gdb.
- Learn advanced tool usage for programs like git and gdb.
- Begin to establish an online portfolio to make yourself attractive to future employers.

Course Structure

Methods of Instruction

- Lecture Integrated with Hands-On Lab Activities
- Independent Programming Assignments
- Term Project
- Exams

Grading

This course is graded using a weighted average among four categories. The assignments within each category are equally weighted and are all graded out of 100 points. Hence your final numeric grade is computed by finding the average of each category, and then multiplying them by the corresponding weight. The weights for each category are as follows:

Category	**Weight**
Exams	20%
Attendance & Hands-On Lab Activities	20%
Programming Assignments	35%
Term Project*	25%

* Note that your course letter grade cannot be higher than the letter grade you earn for the term project.

Grade reports will be returned to you via a private github repository. If you notice any inaccuracy in your grading, please report it as soon as possible.

Letter grades will be assigned according to the following scale:

A+	96.7–100%	B+	86.7–90%	C+	76.7–80%	D+	66.7–70%	F	less than 60 %
A	93.3–96.7%	B	83.3–86.7%	C	73.3–76.7%	D	63.3–66.7%		
A-	90–93.3%	A-	80–83.3%	C-	70–73.3%	D-	60–63.3%		

Assessments

The standards of assessment in each grading category will be as follows.

Exams *(20% of the final grade)*

There will be two exams given in this class: a midterm exam and final exam. The final exam is not comprehensive, it merely covers the material from the second half of the course. Both exams are mixed format exams including multiple choice, fill-in-the-blank, and short answer questions.

Attendance & Hands-On Lab Activities *(20% of the final grade)*

Attendance in this class is mandatory, and attendance is defined as full participation for the entire duration of a class period. Attendance will be taken at the end of every class period. Partial attendance for a class meeting will receive no credit. All assignments will be due at

the beginning of their respective class period and all quizzes will be given during the first few minutes of their respective class periods. Failure to submit an assignment and/or failure to take a quiz will also result in an absence for the day.

In most class meetings, there will be a hands-on activity which we will complete together as a class. Should any part of these assignments not be completed in class, it is your responsibility to complete them on your own. In-class lab assignments are due on the Friday of each week, and are turned in electronically. No late lab submissions will be accepted.

Programming Assignments ***(35% of the final grade)***

There will be five programming assignments given throughout the year. You are expected to work independently on these assignments. The source code you submit must come from one of three sources:

- Your own original code.
- Code from class examples and lab assignments.
- Code from your textbook.

If you do use code from class examples and/or your textbook, you are required to cite the sections of your code that are not original. Using source code from any other source will be considered cheating, and will be dealt with according to the cheating and plagiarism policy stated later in this syllabus. (This includes paraphrasing code found on code repository sites such as github, gitlab, etc. It also includes using code snippets from help sites such as stack overflow. You may use these sites to study, but you must not ever submit code from these sources as your own!)

Term Project ***(25% of the final grade)***

There is a project which is due at the end of the term. The project is an open ended project in that you will create a video game of your chosing. The constraints of this project are:

- The game must be written in C++.
- The game must make use of the FLTK graphics library.
- The game must consist only of source code form the following sources:
  - Your original code.
  - Code from the textbook.
  - Code from labs and in-class coding examples.

  As is the case with the programming assignments, you must cite the source of any code you use in this project. Doing otherwise constitutes plagiarism.
- At the end of the term, you will make the code public under an open source license of your chosing.

This project is assigned as of the beginning of the course, and various milestones will be announced as the semester progresses.

Your overall letter grade for the course cannot be higher than the letter grade on this assignment.

Schedule

This is the tentative schedule for our course. There may be some slight modifications to the following according to the needs of the semester. However, the exam dates are fixed, and will be followed. All exam dates are shown in bold font, and it is absolutely vital to your success in this course that you attend the exam days. The final exam period is set by the registrar. Attendance on the date of the final exam is absolutely mandatory; any student failing to appear on this date will receive a failing grade for the course.

January 2020

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

- **Wed January 8** - Reviewing the Basics
  - Review Chapters 1 and 2
  - Lab Activity: Programming Project P1.2 From Big C++
- **Mon January 13** - C++ Design and Thinking
  - Review Chapters 2 and 3
  - Lab Activity: Programming Project P3.3 From Big C++
- **Wed January 15** - C++ Design and Thinking
  - Review Chapters 4 and 5
  - Lab Activity: Programming Project P5.9 From Big C++
- **Mon January 20** - (MLK Day, no afternoon classes)
- **Wed January 22** - C++ Classes and Objects
  - Review Chapters 8 and 9
  - Lab Activity: Programming Project P9.5 From Big C++
  - Lab Activity: Write a driver program for the battery class.
- **Mon January 27** - Arrays and Pointer Basics
  - Read Chapter 6.1 – 6.6 and 7.1 – 7.4

  - Lab Activity: Programming Project P7.2 from Big C++

- **Wed January 29** - Pointer Basics
  - Read Chapter 7.5 – 7.6
  - Lab Activity: Programming Project P7.1 from Big C++

February 2020

Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

- **Mon February 3** - Objects, Pointers, and Dynamic Allocations
  - Read Chapter 7.8 and 9.10
  - Lab Activity: Programming Problem P9.2 from Big C++
  - Lab Activity: Implement a driver program using pointers and dynamic allocation.
- **Wed February 5** - Objects, Pointers, and Dynamic Allocations
  - Lab Activity: Programming Problem P9.4 from Big C++
  - Lab Activity: Fully implement everything using dynamic allocation.
- **Mon February 10** - An Introduction to FLTK
  - Begin work on Program 1: Pocket Calculator (Due March 2)
  - Read FLTK 1.3.5 Programming Manual sections “Introduction to FLTK” and “ FLTK Basics”
  - Lab Activity: Create the FLTK Hello World Program
  - Lab Activity: Create a Makefile template for FLTK Applications
- **Wed February 12** - An Introduction to FLTK
  - Read FLTK 1.3.5 Programming Manual section “Common Widgets and Attributes”
  - Lab Activity: FTLK Window Layout
- **Mon February 17** - Inheritance and Polymorphism
  - Read Chapter 10
  - Lab Activity: P10.7 from Big C++
- **Wed February 19** - Inheritance, Polymorphism, and Drawing Shapes

  - Read FLTK 1.3.5 Programming Manual section "Drawing Things in FLTK"
  - Lab Activity: Drawing Shapes
- **Mon February 24** - Events and Shapes
  - Read FLTK 1.3.5 Programming Manual section "Handling Events"
  - Lab Activity: Click Art
- **Wed February 26** - Responding to Events
  - Lab Activity: Enhanced Click Art

March 2020

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

- **Mon March 2** - Review
  - Program 1: Pocket Calculaor is Due
- **Wed March 4** - Midterm Exam
- **Mon March 9** - Custom Widgets
  - Read FLTK 1.3.5 Programming Manual section "Adding and Extending Widgets"
  - Begin work on Program 2: Conway's Game of Life (Due March 30)
  - Lab Activity: Big Pixel Grid
- **Wed March 11** Custom Widgets
  - Lab Ativity: Big Pixel Editor
- **Mon March 16** Spring Break
- **Wed March 18** Spring Break
- **Mon March 23** Animations, Games, and Simulations
  - Read the FLTK example `animated.cxx`
  - Get animated.cxx running.
  - Lab Activity: Trajectory
- **Wed March 25** Animations, Games, and Simulations

  - Lab Activity: Sprite Engine Examples

- **Mon March 30** Namespaces, Frameworks, and Alpacas
  - Program 2: Conway's Game of Life is Due
  - Begin Work on Program 3: A Pack of Attack Alpacas (Due April 8)

April 2020

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

- **Wed April 1** - OOP Backtracking Algorithm
  - Read Chapter 11.6
  - Lab Activity: Sudoku Solver
- **Mon April 6** - OOP Backtracking Algorithm
  - Lab Activity: Insanity
- **Wed April 8** - Advanced Use of Debuggers
  - Program 3: A Pack of Attack Alapacas is Due
  - Begin Work on Program 4: Pegboard Games (Due April 22)
- **Mon April 13** - Machine Code and C
  - Read https://learnxinyminutes.com/docs/c/
  - Lab Activity: Disassembling a Programming in gdb
- **Wed April 15** - The UNIX Memory Model
  - Lab Activity: Exploring the Stack
- **Mon April 20** - Smashing the Stack for Fun and Profit
  - Read: Smashing the Stack for Fun and Profit
  - Lab Activity: Old School Virus Writing
- **Wed April 22** - Game Presentations & Review
  - Program 4: Pegboard Games is Due
  - Activity: Demonstrate and Play Games
- **Tue April 28 3:30PM** - Final Exam

Course Policies

Late Policy

No late work will be accepted under any circumstances (except as mercy and decency may dictate in extremely rare events).

Extra Credit

No extra credit will be given under any circumstances.

Excused Absences

In some cases, absences may be excused. These include:

- School Sanctioned Events (Sports, Concerts, etc.)
- Severe Illness
- Family Emergencies
- Court Appearance / Jury Duty

In the case of a school event, notice must be given at least one week prior to the absence. The notice must include a signed note from the faculty or staff member in charge of the event. This note must be given in physical form, electronic notes will not be accepted. In the case of illness, a doctor's note is required. Note that except in extreme circumstances, doctor's appointments do not qualify as a valid reason to miss a class. Please be respectful of the other students, and schedule appointments during your free time.

Family emergencies will require some form of proof. Where possible, you must give advance notice of missing a class. The exception to this would need to be fairly severe, and hopefully it will not come up. For court appearances and/or jury duty, you must provide a copy of your summons. You may redact any details you wish, save for the actual date and time of your appearance. Court appearances must be cleared at least one week in advance.

Making Up Excused Absences

Should you be in a situation in which you receive an excused absence, this in no way will extend your due dates (excepting extreme emergencies). You must make up any test at a designated time prior to your excused absence. Also, homework or projects must be submitted prior to the class period in which they are due.

Communication and Extra Help

You are always welcome at office hours for help with any questions you may have about the course. For help at other times during the day, stop by or call my office to see if I'm available. You can also contact me by email, but often I can better help you face to face and may respond with a request that you come to see me. Note that I do not typically respond to email between 5 p.m. and 8 a.m. You may make appointments to see me at other times if your schedule does not permit you to attend my office hours.

Plagiarism and Cheating

You are expected to do your own work. Never submit the work of others, never give unauthorized assistance to others, do not use unauthorized aids during exams, and do not ask for help from other faculty members without the approval of your professor. Plagiarism and cheating are serious offenses that will not be tolerated. Explanations regarding these offenses and how they are handled can be found in the MC Student Handbook at
https://www.maryvillecollege.edu/academics/catalog/handbook/section-nine/.
You are expected to read and understand these policies. Offenses on specific assignments, quizzes, or exams will result in a score of 0 on the relevant assignment, and a letter of censure will be placed in your college file. Repeat offenses will result in further disciplinary action, including the possibility of failing the course.

Students with Disabilities

Any student who feels s/he may need learning or physical accommodation(s) based on the impact of a disability should contact Services for Students with Disabilities to discuss your specific needs. Please contact 981-8124 to coordinate reasonable accommodations for students with documented disabilities. The Disability Services office is located in the Learning Center in the basement of Thaw Hall. Undocumented disabilities will not be accommodated.