CSC 1820
Principles of Computer Science II
Dr. Mark Mahoney
Spring 2022
MWF 1:35pm-2:40pm
MWF 2:50pm-3:55pm
office: (262)551-5869
mmahoney@carthage.edu
Office Hours:
DSC 192
TR 2:15pm-4:00pm
MW 12:30pm-1:15pm

Textbook:
An Animated Introduction to Programming in C++ (this is a free 'book' of guided code walk-throughs written by me)

Grading Criteria:
Programming Assignments 100%

Course Description:
This is the second course in computer programming that you will be taking at Carthage College. As you may have realized from taking CS1810 computer programming is a very rewarding, demanding, exhilarating, time consuming and sometimes incredibly frustrating science.

We will continue to use the C++ programming language. However, this course is not about learning the fundamental syntax of the language. Rather, it is about learning how to apply those constructs from CS1810 to solve more complex real world problems. We will spend almost every session discussing a problem and I'll ask you to come up with solutions. We will then discuss the pros and cons of the solutions. Then you will be asked to implement those solutions in the homework. I am expecting significant participation in the discussions we have about the problems.

In addition to problem solving, we will be learning about a new way to think about writing programs called Object Oriented Programming. OOP is a fundamentally different type of programming that builds on the procedural programming techniques we have learned in CS1810. There is some new syntax that we will need to learn to write OO programs but it is relatively straightforward. OOP is the dominant software design paradigm used in industry today.

As some of you will soon find out, programming is not for everyone. You must have excellent organizational skills to avoid common errors. It is important to realize that each and everyone of you will make errors when writing programs. The people who succeed are those that do not give up easily, those who can spot their own errors quickly and correct them. On a side note, at some point you may feel like you are way behind other students. I have found this to be mostly untrue. Every single student struggles with programming- you are not alone! The students who overcome this feeling are the ones who seek out help early and often. Having said this, however, not liking programming does not preclude you from an interesting challenging career in the field of computer science or information systems. There are many great jobs out there that don't involve programming in C++.

Now is a wonderful time to be a computer programmer. Computer science is consistently ranked as one of the majors with most opportunities after graduation. The Bureau of Labor and Statistics projects that people with degrees in computer science will earn more and have more job opportunities than most other fields. In addition, “Employment of software developers is projected to grow 22 percent from 2012 to 2022, much faster than the average for all occupations.”

Now is also a great time to be a entrepreneur with computer science skills. It literally takes almost no capital to start a software based business. If you have a good idea and some time to build the software to implement it, you can start your business while in college. Google, Microsoft, and Facebook were all started by college students.

In a recent article by Marc Adreessen called, ‘Why Software Is Eating The World’ the author describes the upcoming disruption of just about every business sector by the software industry, “Six decades into the

computer revolution, four decades since the invention of the microprocessor, and two decades into the rise of the modern Internet, all of the technology required to transform industries through software finally works and can be widely delivered at global scale." No matter what your interests are, knowing how to program a computer will be critical in the next 50 years that you will be working.

Objectives:
• Develop strong problem solving skills.
• Develop an ability to analyze a set of requirements, come up with a design that solves the problem, and then be able to translate that design into an implementation.
• Develop a firm understanding of object-oriented concepts in C++.
• Further develop the student's ability to use a C++ editor, compiler, and debugger.

Class/General:
This is a difficult subject that requires reading and practice. If you do not do both, you will not pass this course. Programming courses in general require a great deal more effort outside of class on the student's part than most other courses. This class will be no exception. You cannot learn how to program simply from watching me do it, just like you could not expect to learn how to paint from watching a more experienced painter. Writing software is an art and a craft. My main goal is that you learn how to become a creative problem solver! I expect each student to come fully prepared to participate in class discussions, ask questions and see me outside of class for help if necessary.

Class typically starts with some review of the previous material. Then, a discussion will follow from the day/week's assigned problems. I will step through my code several times during each class. Feel free to ask for help after class. Also, feel free to see me in my office. Students can email me with any questions but face to face questions are more likely to be answered easily. There are no dumb questions!

Class attendance is required. Repeated unexcused absences will negatively affect your grade.

Computers/Computer Lab:
You must use C++ outside of class. I encourage you to bring a laptop computer into class if you have one. However, do not surf the web while you should be programming. It would be nice if you have a C++ compiler installed on your personal computer. I will provide a link to a free C++ compiler that you can install on your own machine (Windows Visual Studio Community Edition or Apple's XCode). If you don't have a personal computer than you can use there are sites like repl.it and cpp.sh that will work in a pinch.

Class Web Site:
I have created a website for this class to better distribute information. The website will provide each student with access to his or her grades, class announcements and other resources.

Programming Assignments:
In order to learn the C++ language you must practice outside of class several times a week. Simply attending class is not sufficient to learn any programming language, especially one as complicated as C++. There will be several programming assignments. Check the homework page for due dates.

All assignments must be submitted using schoology. I do not accept late homework but I do give partial credit for programs that do not work completely. Please upload your source code files individually using the digital drop box. This semester every student will get one 'no questions asked' three day extension and one 'no questions asked' seven day extension for the assignments.

Read the requirements for each assignment carefully. Ask questions on any portion that is not clear.

Students are encouraged to help each other while learning the material. The exchange of ideas enhances the learning process. However, the final product must be your own. Plagiarism of assignments and cheating on exams are not permitted. Any student caught doing so will automatically receive a failing grade. You can find the academic honesty guidelines here (https://www.carthage.edu/community-code/academic-concerns/academic-honesty-guidelines/) and the consequences for violating them here (https://www.carthage.edu/community-code/academic-concerns/penalties/).

Carthage College strives to make all learning experiences as accessible as possible. If you anticipate or experience academic barriers due to your disability (including mental health, learning disorders and chronic medical conditions), please let me know immediately so that we can privately discuss options. To

establish reasonable accommodations, you also need to register with Diane Schowalter in Learning Accessibility Services (dschowalter1@carthage.edu).