CS 112 – Computer Science Fundamentals 2
Exam #2 – Study Guide

Schedule

Exam #1 will be given on **Tuesday, November 17, 2020. You may take the exam online in any 80-minute period between 11:00 AM and 5:00 PM. The instructor will be available for questions via Zoom ONLY from 11:00 AM to 12:20 PM, during the normal class time.** The exam is 80 minutes (+ accommodations).

Administrivia

This is an exam given online via Canvas. As such, there is no effective way to limit the resources you have available during the exam. I will, though, write the exam with the idea that you have a sheet of notes that you have prepared in advance. I will even make suggestions below about what that sheet of notes should have on it. But you can add whatever you think is useful to that sheet, and you are not required to turn it in. Hold onto that sheet, though, as the final exam will cover all material in the class and you can use it then, too!

Keep in mind that the exam will be timed in a manner that does not allow extensive referrals to external materials. It will be geared towards testing your familiarity with the presented materials in class and what we've covered in the textbook. This is another good reason to condense your notes to a single "cheat sheet" you can use quickly and easily to keep pace with the problems in the time provided.

The exam will be written in a way that having a single letter-size sheet of notes is best.

Format of exam

The exam will be in a format similar to one that would be given on paper in a classroom. To that end, it is NOT expected that you will enter any code into an IDE to compile and run it, and there will in all likelihood NOT be enough time for you to do so!

Among the types of questions you may be asked include:

- Examine existing fragments of C++ code, and answer questions about the code (what value would an identifier have, what line of code would be added to do a particular thing, etc.)
- Supply single lines (or a few lines) of C++ code to perform a particular thing
- Write fragments of C++ code that show the correct use of code structures, input/output using cin/cout and file I/O
- Write expressions in C++ that match a given description or that would be used in C++ code
- Define functions (write code for header and body) in C++ that expects a particular set of arguments, performs a particular task, returns a computed value, and has certain specified side effects
- Examine code structures (such as while loops, for loops, etc.) and discern what those structures do, how many times they iterate, write a predicate (Boolean) expression that would be an effective sentinel for the loop's execution, etc.
- Given a fragment of C++ code, write out the exact output to the screen the code would generate
- Reading and writing code that makes use of C++ methods and data members inside them

Academic Honesty

It is ABSOLUTELY required that the work you present on the exam is your own! No real-time collaboration with other students is permitted. Canvas has features to combat cheating, such as randomizing the order of questions, randomizing the listing of multiple choices on questions, and even offering students different sets of questions from a "test bank" so that no two tests are exactly alike!

Material to be Covered

You are responsible for all material covered in class. The Savitch text is very comprehensive, to references below to chapters in the text are there just to point out where in the text they are. You will NOT be responsible for all information in those chapters, just the parts we've covered in lectures, labs, and assignments. **In addition, material covered in Exam #1 will be used as needed in Exam #2 also!**

Chapter 9 – Pointers, Dynamic Allocation, and Dynamic Arrays

- What is a pointer? What does it do? How can it be used?
- How do you declare a pointer variable in C++? What are some of the ways you can initialize it? Whats are some of the ways you can set it?
- According to class style standards, if a pointer is currently not "pointing" anywhere, then to what special value should it be set? NOTE: both **NULL** and **nullptr** will be accepted as a correct answer to this!]
- How can you make a pointer point to something?
- How can you change what a pointer points to?
- How can you access the contents of what a pointer points to?
- If a pointer is pointing to an object, how is -> used to call the methods associated with the object that the pointer is pointing to?
- How how can you dynamically allocate memory and have a pointer point to that memory? You should be able to do this for single values (like int, double, bool, etc.), arrays, and objects.
- How do you free (or deallocate) this memory when you are done using it? What is the special syntax needed if you are doing this for a dynamically-allocated array?

Chapters 10&11 – C++ Classes, Methods, Data Members, and Overloading

- NOTE: C++ structs will NOT be on the exam!
- You should be comfortable with defining and implementing a C++ class, and with reading and using C++ classes. You should be able to write a class definition and also implement that class, given specifications of what is desired.
- Given a C++ class, you should know how to declare instances of that class (also known as objects). You should be able to call methods of the class using either object identifiers or pointers to an object.
- What is the purpose behind having a public part and a private part of a class definition? What is normally placed in the public part of the the class definition? In the private part of a class definition?
- What is a class data field (also called a class member variable?)
- What is a class method (also called a class member function)?

- You should be familiar with methods, and specifically the following types of methods: Constructors, Zero-Argument Constructors, Copy Constructors, Destructors, Accessor Methods, Mutator Methods, Overloading Operator Methods (which are mentioned in Section 11.2 of the text, not in Chapter 10!)
- You should know how to write a line of code that uses each of these types of methods, and especially be aware of how a zero-argument constructor is called.
- How do you write a function that expects an object (or a pointer to an object) as a parameter?
- What does overloading a method within a class mean? How do you overload a method within a class?
- Why is it considered common practice to overload constructors in a class?
- What is the purpose of an accessor/getter method? What is the typical visibility for these methods?
- What is the purpose of a mutator/setter method? What is the typical visibility for these methods?
- What is the scope resolution operator **::** used for when implementing a class's methods?
- What is a private method? When would you want to define one?

Given a class definition, you should be able to use that class by declaring instances of it (using both zero-argument and non-zero-argument constructors), calling its accessor methods, mutator methods, and calling its "other" methods, as needed.

Section 10.4, Section 15.1 – Derived Classes and Inheritance

- What is meant by a base class? What is a derived class, and how is it related to a base class?
- What is the syntax for connecting a derived class with a base class?
- What does a derived class inherit from a base class? What does a derived class NOT inherit from a base class?
- What is the protected section of a base class's definition? How does it differ from the private section?
- What is meant by a redefined method, and how is that different from an overloaded method?
- What is the syntax for specifying exactly which class to use when calling a method?
- What is the relationship between an object in a derived class and the derived class's base class?
- Can a base class method be called using a derived class object?
- Can a derived class method be called using a base class object?

The following will NOT be covered in Exam #2

- Polymorphism and Virtual Methods
- Multiple Inheritance
- Abstract Classes
- C++ Vectors