Topics to Study for Exam I

Below are the learning outcomes from each of the four modules we have covered leading up to Exam I. You should be able to answer each question or perform the action listed. Feel free to ask questions on Piazza if a topic is unclear.

M1. Primitive Data Types, Variables, & Arithmetic Operators
LO1. List C++ primitive data types and explain the appropriate use of each data type
LO2. List & identify C++ arithmetic operators, translate math equations to C++, and solve arithmetic expressions
LO3. Construct a simple interactive application
LO4. Describe how a computer generates a program from code
LO5. Explain where and how variables are stored in memory
LO6. Discuss how to convert one data type to another
LO7. Explain how a computer generates random numbers and write a program that generates random numbers
LO8. Create a program with formatted output.

M2. Boolean Expressions & Conditionals
LO1. Identify C++ control structures and conclude which branch a sample program will execute
LO2. List C++ logic operators and evaluate Boolean expressions consisting of multiple logic operators
LO3. Evaluate the resultant output of a code block containing a control structure.
LO4. Generate pseudocode to solve a decision problem
LO5. Write a program that implements the pseudocode and solves the problem
LO6. Identify common errors exhibited in code that contains Boolean expressions. Correct errors in Boolean logic that causes the program to not execute as intended.

M3. Looping (for / while / do-while)
LO1. Identify C++ repetition structures and explain the following terms: looping parameter, stopping condition, and looping parameter modification.
LO2. Explain the appropriate use and differences between a while loop, for loop, and a do-while loop.
LO3. Convert a program written with a for loop to a program using a while loop and vice versa
LO4. Explain how scope relates to looping structures
LO5. Evaluate the resultant output of a given code block containing a loop.
LO6. Generate pseudocode to solve a repetition problem
LO7. Write a program that implements the pseudocode and solves the problem
LO8. Identify and correct errors in looping structure and program logic.
LO9. Discuss the effect and purpose of using break and continue in a loop.
LO10. Solve an expression containing a prefix or postfix increment operation.

M4. Arrays & Algorithms
LO1. Describe the differences between & advantages of an array & vector.
LO2. Describe the differences between & advantages of a c-style string & a string.
LO3. Sketch how an array is stored in memory denoting the base address and element step size
LO4. Sketch how a multidimensional array is stored in memory denoting the base address, row step size, and element step size
LO5. Explain how sorting an array affects the performance of searching for a value in an array
LO6. Evaluate the resultant output of a given code block containing an array, string, or vector.
LO7. Construct a program that accesses an element in a vector/string, accesses an element in a multidimensional array, modifies an element in a vector/string, returns the length of a vector/string, changes the length of the vector/string, and other vector/string operations
LO8. Generate pseudocode to (1) find the minimum & maximum value in an array (2) sort an array using selection sort
LO9. Write a program that implements the corresponding pseudocode using an array
LO10. Identify and correct errors in array structures, element access & modification, and program logic.
LO11. Create string variables and apply string functions to access or modify the string. (length, at, find, substr)
LO12. Create vector variables and apply vector functions to access or modify the vector (size, at, push_back)