**CSE-278: Introduction to Systems Programming**

**Homework #2**

**Due: Wed Feb 13 2019 before 11:59 PM**

**Email-based help Cutoff: 5:00 PM on Tue, Feb 12 2019**

Maximum Points For This Part: 22

|  |
| --- |
| **Submission Instructions**  This homework assignment must be turned-in electronically via Canvas. For this part of the homework you should supply answers to each question in this document. Prior to commencing work on this part, rename this MS-Word document using the naming convention MUid\_hw2.docx (example: raodm\_hw2.docx). Once you have completed answering the questions save this document as a PDF file (don’t just rename the document; that is not the correct way to save as PDF) and upload it to Canvas with C++ source. **Do not submit archive files such as: zip,7zip,rar,tar,tar.gz,rpm,deb etc.**    **Copy pasting from online resources is plagiarism. Instead you should understand concepts and explain them using your own words!** |

|  |
| --- |
| **Objective** |
| The objective of this part of the homework is to:   1. Review the basic C++ concepts 2. Practice developing C++ programs similar to questions that will appear on job interviews & exams. |

|  |  |
| --- | --- |
| **Name:** | Nandini Agrawal |

**Required reading**

Prior to answering the questions in this homework briefly review the following chapters from the E-textbook (link in Syllabus on Canvas) “C++ How to Program” (all students have free access to the electronic book):

|  |  |
| --- | --- |
|  Chapter 1 (Introduction) |  Chapter 2 (Expressions & I/O) |
|  Chapter 3.1, 3.2 (String & methods) |  Chapter 4 (if-statements) |
|  Chapter 5.1 – 5.10 (loops & switch) |  Chapter 6 (functions/methods) |

|  |  |
| --- | --- |
|  | Although the Safari E-books are available to all students there are only a limited number of concurrent licenses to access the books. Consequently, do not procrastinate working on this homework or you may not be able to access the E-books due to other users using them. |

1. Briefly (1 sentence each) state 3 advantages of using the C++ standard library (as inferred from Chapter 1.10)? [**1 points**]

|  |
| --- |
| 1. Does not have any middle person such as JVM. 2. It not only specifies the syntax and semantics of generic algorithms, but also places requirements on their performance. 3. Easy to understand and read. |

1. Briefly (about 2 sentences each) discuss 2 key philosophical differences between C++ and Java [**2 points**]

|  |
| --- |
| C++ is platform dependent but java is platform independent.  C++ supports multiple inheritance but java doesn’t support multiple inheritance through class. |

1. Briefly (1 to 2 sentences for each phase) describe the 6 phases involved in creating and running a C++ program (see Chapter 1.9) [**3 points**]

|  |
| --- |
| 1. Editing a program- to make any necessary corrections and then save the program. 2. Preprocessing a C++ Program- give command to compile the program. 3. Compiling a C++ program- compiler translates the C++ program into object code. 4. Linking- links the object code with the code for the missing functions to produce an executable program. 5. Loading- takes executable image from disk and transfers it to memory. 6. Execution- executes the program one instruction at a time. |

1. What is the difference between the Internet and the World Wide Web (WWW) as described in Chapter 1.12 [**1 point**]?

|  |
| --- |
| Internet is a network of networks that connects different networks together to standardize various protocols whereas World Wide Web is a collection of hardware and software associated with the internet that allows computer users to locate and view various documents on almost any subject. |

1. Briefly (2 or 3 sentences) describe code “refactoring” and provide an example (see Chapter 1.13) [**1 point**]?

|  |
| --- |
| Refactoring means going through the program again to make them clearer and easier to maintain while maintaining the correctness and functionality. Ex: writing methods with only 25 characters. |

1. What is the significance of the main function returning zero? Can it return some other value (Chapter 2.2.)? [**1 point**]

|  |
| --- |
| To indicate that the program ended successfully. |

1. What is the use of the using namespace std; declaration in a C++ program (Chapter 2.7)? [**1 point**]

|  |
| --- |
| Std:: is required before cout when we use names that we’ve brought into the program by the preprocessing directive #include <iostream>. It specifies that we are using a name. |

1. What is an unsigned numeric data type? State 2 purposes for using unsigned numbers? [**2 point**]

|  |
| --- |
| It is a variable that can only hold positive numbers or zero. We can use it while writing programs where we can’t have negative values as inputs like money or number of people etc. It is used to put the values to valid ranges. |

1. What is the size\_type or size\_t data type? [**1 point**]

|  |
| --- |
| Can store the max size of an object including array. It has a size of 8 bytes and range is from 0 to +4 trillion. |

1. What is pseudocode? Where/why would you use pseudocode within (or as part of) the source code of a C++ program? [**1 point**]

|  |
| --- |
| Pseudocode is a description of what the program should look like but it uses normal formal language instead of using a programming language. It can be used in the source code in the comments to tell how the program needs to be implemented. |

1. Review the source code in Figure 5.11 (Chapter 5.9). Briefly describe what input(s) will be needed to terminate the while-loop on line #24 under Linux? [**1 point**]

|  |
| --- |
| User input will go into the variable grade. To terminate the code cin << grade. |

1. What is a function prototype and when is it needed (Chapter 6.4)? [**1 point**]

|  |
| --- |
| A function prototype describes the max function without revealing its implementation. It is declaration of a function that tells the compiler the function’s name, its return type and the types of its parameters. It ensures that the class is defined before main creates and manipulates objects of that class. |

1. What is a function-call stack? How can you observe the call stack in the IDE used for your course? (Chapter 6.11)? [**1 point**]

|  |
| --- |
| It supports the function call/ return statements. It also supports the creation, maintenance and destruction of each called function’s local variables. The activation record tells main how to return to the operating system. |

1. Given a constant string object str (const std::string str) what methods can be called on str? Give an example of 1 method that can and 1 method that cannot be called on the object str? [**2 points**]

|  |
| --- |
| Constant method can be called on str.  const char [4]) const'  str.replace(10, 5, "red");  Replace method cannot be called on str. |

1. What is the advantage of using constant reference parameters in functions (Chapter 6.13)? [**1 point**]

|  |
| --- |
| When the member function is called it receives as copy of its string argument. String objects can be large, so the copy operation degrades an application’s performance. That’s why they should be passed to functions by reference. |

1. What is a common programming error that occurs when returning reference to a local variable (Chapter 6.13)? [**1 point**]

|  |
| --- |
| Unless the variable is declared static the reference refers to a variable that’s discarded when the function terminates. |

1. When is it safe/appropriate to return a reference from a method? [**1 point**]

|  |
| --- |
|  |