CS&E 1222

Lab 1 – C++ Basics Lab Assignment – 20 points

- ✓ The *lab* must be accomplished solely by you:
  - ➢ DO NOT look at anyone's code other than your own, including code from another's student in your section or another section of the course, or any third party source, e.g. the Internet
  - ➢ DO NOT share or copy anyone else's code for any graded assignment
  - ➢ DO NOT work in pairs or groups
- ✓ All suspected cases of academic misconduct will be reported to the University.

Setting up the Programming Environment

1. Study course material before you start this lab including the textbook readings, your lecture notes, and course notes found on Carmen. Especially study lectures 2 (Introduction) and 3 (Variables and C++ Data Types) of the course notes. Also study the Powerpoint slides *Unix Tutorial* found on your Carmen account under the "Content" tab and then the "Tutorials" module.

2. Login to your Linux account.

3. At the Linux command line type `mkdir lab1`. This will create a new directory named **`lab1`**. Work out of this directory. In order to do that, type `cd lab1`. This changes the current working directory to the directory **`lab1`**.

4. If you have created the directory **`lab1`**, then just type `cd lab1`.

5. Copy the file **`basics.exe`** in the directory **`/class/cse1222/9643/lab1`** by typing

```
cp /class/cse1222/9643/lab1/basics.exe .
```

Be sure to include ***9643*** (this is your course section indicator) and the period ".".

Programming Assignment

1. Create a new file called **`basics.cpp`** using the Emacs editor (as you did in lab last week) which will contain a program that asks the user to enter values of different types and then displays computation performed on these values.

Enter the following initial code in the file **`basics.cpp`**:

```
// File: basics.cpp
// Created by: Your Name
// Created on: Creation Date

/* This program illustrates the use of keyboard input, data
   types, variables, assignment statements, and output
*/

#include <iostream>
using namespace std;

int main()
{
  double d;
  int i1, i2;
  char c;

  // insert your solution here


  return(0);
}
```

2. Replace "Your name" with your name.
3. Replace "Creation Date" with today's date.
4. The program above declares four variables: a character value, a double value, and two integer values. You will write a program that will prompt the user (use `cout`) and read (use `cin`) these four values from the keyboard and then displays computation on these values. Your program will first ask the user to enter a character value, then a double value, and finally the two integer values. It will then compute and display the results of various math formulas.
    a. From the Linux command line run the solution I provide in the executable file **`basics.exe`**.
    b. Your solution must match the behavior and output of this solution **exactly**. This includes the text displayed, empty lines, and results of the computation displayed to the screen.
    c. Run the solution a few times with different input values to get a good idea of what computations are being done.
5. Write your solution in the above initial code below "// insert your solution here" so that your program behaves exactly as the solution provided in the executable **`basics.exe`**. Your program must produce exactly the same output as the solution **`basics.exe`** on the same inputs values.
    a. Do not use any more variables in your solution than the four provided above. You must use all four of these variables in your solution.

b. ****** Do not write your code all at once!*** *** Instead write your code ***incrementally***. This means write only a couple or few lines of code in your solution and then compile and run the code immediately. Test this code with different input values. If these few lines of code work, then write a few more lines and do the same over again. By writing your solution step-by-step and repeatedly compiling, running, and testing as you develop your program you will arrive at a correct solution quicker. It is good to develop good programming habits early (ask your instructor).

   For example, first write only enough code to simply prompt the user for a character "`Enter a character: `" and then read a character from the user. The entered value should be stored in the variable `character` (see code template above). If this code compiles and runs successfully then continue with your solution by writing only enough code to prompt and read from the user for the double value. If your code compiles and runs successfully for both of these tasks then continue to write your solution ***incrementally*** in this manner.

Compile and run your program. Thoroughly ***test*** your program by checking its output against **`basics.exe`**. Run your solution on **many** input values to ensure that all mathematical formulas are correct.

Submit Your Work

Important: Any program which does not compile and run will receive no credit!
If you are not sure what this means please ask your instructor.

Submit your file **`basics.cpp`** using the *Lab1* drop box on Carmen under the "Assignments" tab. **DO NOT** submit the file **`a.out`**. **DO NOT** submit uncompleted work from *Quiz1*. This will not be graded.