CS 271
Lab 5

Chapter 15 - Introduction to C++
Chapter 16 - Introduction to Classes

40 points total

All Programs Must Have:

Header Comments and Inline Comments
Consistent Indentation and Spacing

All programs must compile. Programs that do not compile will receive a grade of zero.

C++ : This lab is to be written in C++.

NO printf or scanf
NO #include <stdio.h>
and no other C libraries in the #include statements either
NO \n in print statements

YES **#include <iostream>;** and **#include <iomanip>;** and **using namespace std;**
YES cin and cout
YES you can declare the loop control variable in a for loop header
YES use endl to output a newline

The C++ compiler is called g++.

C++ source code files end with cpp.

makefiles work the same way as for C except that you use g++ in the compiler commands instead of gcc.

The main function header is slightly different in C++.

```
int main( )          // no void in the parameter list
```

Getting Input and Printing Output in C++

Input C++ Style

cin is the name of the input "stream" in C++ and >> is called the "stream extraction operator". You can use cin to input numbers very easily.

```
int x;
cin >> x;
```

You can use cin to input strings that don't have spaces in them.

```
string word;
cin >> word;
```

The >> points like an arrow from cin to the variable name.

You can input more than one variable in the same statement.

```
int a, b, c;
cin >> a >> b >> c;
```

Output C++ Style

cout is the name of the output "stream" in C++ and << is called the "stream insertion operator". You can use cout to output just about anything in C++

```
int x = 10;
cout << x;
```

```
string sentence = "Today is a good day.";
cout <<  sentence;
```

The << points like an arrow from the variable name to cout.

You can output more than one variable in the same statement.

```
int a = 4, b = 7, c = 1;
cout << a << b << c;
```

To output a newline,

```
cout << endl;
```

Part 1: C++ Basics

- **Open up your editor and type in the following C++ program.**

```
#include <iostream>
using namespace std;

// This program inputs width, height, and length of a box then
// outputs the volume of the box.

int main()
{
   int width, height, length, volume;

   cout << "Enter the width of the box: ";

   cin >> width;

   cout << "Enter the height of the box: ";

   cin >> height;

   cout << "Enter the length of the box: ";

   cin >> length;

   volume = width * height * length;

   cout << endl << endl << "The volume of a box with length " << length
        << ", width " << width << " and height " << height << " is: "
        << volume << "." << endl;

} // end main
```

- **Save the program with the name volume.cpp.**

- **Compile the program with**

```
g++ -o volume volume.cpp
```

- **Debug if needed.**

- **When you have no syntax errors, run the program with**

```
./volume
```

- **Close volume.cpp.**

- **Start a new file. (You may look at volume.cpp but don't use copy and paste.)**

- **Type the #include and using statements.**

Note: you'll need these preprocessor directives at the top of your program

```
#include <cstring>
#include <iostream>
#include <iomanip>
```

- **Insert prototypes for the following two functions.**

1. Write a C++ function that will accept an array of double values and an integer that contains the size of the array. Without modifying the array AND without creating any more arrays, the function should print out the two highest values in the array. (You may assume that the array has at least two elements.)

   Here's the header of the function:

   ```
   void highTwo ( double array[ ], int size ) {
   ```

2. Complete the following function so that it will accept a pointer to the first character in a null-terminated character string. The function should determine the length of the longest word in the string.

   Words are separated by one space. You may assume that there are only letters and spaces in the string. Also, there are no extra spaces at the beginning or the end of the string and there is at least one word in the string.

   Example: longestWord("This is a difficult question") should return 9.

   The function begins this way:

   ```
   int longestWord ( char *cPtr ) {
      int length = 0, longestLength = 0;
      for (int index = 0; index <= strlen(cPtr); index++ )   {
   ```

- **Write the main function.**

The main function should thoroughly test the highTwo function and the longestWord function.

Allow the user to type in the values for the double array then call the highTwo function.

Allow the user to type in the sentence then call the longestWord function and print the value returned from it.

- **After the main, write the function implementations.**

- **Save the program as myprogram.cpp.**

- **Compile, debug, test. Repeat...**

Part 2: Intro to C++ Classes

1. Download the files BankAccount.h and BankAccount.cpp. Read the code in BankAccount.h carefully so that you know what functions you can call.

2. In the same folder where you stored BankAccount.h and BankAccount.cpp, create a program called lab5.cpp.

   a) Include libraries: iostream, iomanip, and the header file "BankAccount.h"

   b) Write a main function that will create two BankAccount objects and test the member functions of the BankAccount class.

   Note: the syntax for instantiating objects in C++ is different from that of Java. Here's the C++ syntax:

   ```
   BankAccount joesChecking( "Joe", "Smith", 1234 );
   ```

3. Create a makefile. Be sure to use the g++ compiler command.

4. In BankAccount.h add prototypes for two new functions:

   1) a default constructor

   ```
   BankAccount ();
   ```
   -- set the first and last names to the empty string and set the account number to zero.

   2) a member function that will display the data members

   ```
   display( )
   ```
   -- print the data members in this format

   ```
   Customer:  Joe Smith
   Account:   1234
   ```

5. In BankAccount.cpp add the implementations of these two functions.

DO NOT, UNDER ANY CIRCUMSTANCES, USE THE KEYWORD "this".

   3) Add tests to your main function. Use the makefile to compile all programs and create an executable. Test and debug as needed.

Submit a .zip file containing myprogram.cpp, BankAccount.h, BankAccount.cpp, lab5.cpp, and makefile.