OOPS Lab

Programming Assignment № 1

C++ Basics

Saif Ali, CSE Department, JMI 22/08/2023

Read carefully before you begin:

- Total Marks: 30. Each question carries 10 marks.
- You have 2 hours to complete the assignment. Failure to have your program evaluated before you leave the lab will cause forfeiture of the grade for that lab.
- In order to receive full marks, you must demo the full working code and show the output and given an explanation of your approach where applicable.
- Please **save your code** throughout the semester in a place where you do not lose it. You will be required to submit it at the end.
- Use proper filenaming conventions and commenting. Code that is hard to read or understand will incur a penalty.
- Collaboration must kept to general discussions only. Please do NOT share code or directly share answers with each other. Plagiarism is unacceptable.

Problem 1 (10 marks)

Read this tutorial on compilers `https://cplusplus.com/doc/tutorial/introduction/`. Select any compiler that you are comfortable with to compile, build and run your C++ programs. Pick your preferred IDE and make sure it is installed on your system. There are many options. VisualStudio, XCode, gcc, Sublime Text, CodeBlocks etc. Try and stick to the same development environment as far as possible. Write, build and run a simple "HelloWorld.cpp" C++ program that just prints "Hello World".

Hint: Command to build a C++ program using GCC: **g++ hello.cpp**

Listing 1: Sample C++ code – a simple C++ template. (Source: MIT)

```
1 #include <iostream>
2 using namespace std;
3 int main ()
4 {
5 cout << "Hello World!" << endl;
6 return 0;
7 }
```

Problem 2 (10 marks)

Examine the following C++ code listing. Notice the difference from a C program. What does the program do? Using this as a template, **write a C++ program** (not a C program) to convert weight in pounds to weight in kilograms or the other way round. Please use the cin and cout stream objects to take inputs and write outputs. Save your file with a ".cpp" extension. Compile and run your program and examine the output.

Listing 2: Sample C++ code – Convert temperature reading from Celsius to Farenheit or the other way round. (Source: NTU)

```
1  /*
2   * Convert temperature between Celsius and Fahrenheit
3   * (ConvertTemperature.cpp)
4   */
5  #include <iostream>
6  using namespace std;
7
8  int main() {
9     double celsius, fahrenheit;
10
11    cout << "Enter the temperature in celsius: ";
12    cin >> celsius;
13    fahrenheit = celsius * 9.0 / 5.0 + 32.0;
14    cout << celsius << " degree C is " << fahrenheit << " degree F." << ↩
         endl << endl;
15
16    cout << "Enter the temperature in fahrenheit: ";
17    cin >> fahrenheit;
18    celsius =  (fahrenheit - 32.0) * 5.0 / 9.0;
19    cout << fahrenheit << " degree F is " << celsius << " degree C." << ↩
         endl;
20
21    return 0;
22 }
```

Problem 3 (10 marks)

Examine the definition of a class called "Clock" below. Fill in the code for the member functions **setClock** (set clock data), **displayStandard** (display the time in standard format) and **displayMilitary** (display in military or 24 hour format). Create an object (or instance) of this class and set the data using some made up values. Then display in both formats.

Listing 3: Sample C++ code – a simple class definition. (Source: MTSU)

```
#include <iostream>  // this is needed because of
using namespace std; // the "cout" below.

//The Clock class definition starts here.

class Clock
{
  private:
    //declarations of data members that are private
    int hour;          //an hour in the range 1 - 12
    int minute,        //a minute in the range 0 - 59
    int second;        //a second in the range 0 - 59
    string meridian;   //is the time AM or PM

  public:
    // publically accessible methods, i.e., the public interface

    //Set the clock to the specified time
    void setClock (int h, int m, int s, string mer)
    {
    }

    //Display the time in standard notation
    void displayStandard()
    {
    }

    //Display the time in military notation
    void displayMilitary()
    {
    }

};
//The Clock class definition ends with the closing brace and semicolon.
```