CPE 150 Laboratory 1: Introduction to C++ Programming I

Department of Computer Engineering
Yarmouk University

Summer 2017

1 Objectives

- To become familiar with the login process and the C++ environment used in the lab.
- To understand the basics of program design and algorithm development.
- To learn, recognize and correct the three types of computer errors: syntax errors, run time errors and logic errors.
- To learn the basics of an editor and compiler and be able to compile and run existing programs.
- To enter code and run a simple program from scratch

2 Lab Exercise 1 - Opening, Compiling and Running Your First Program

Exercise 1: Logon to your system based on your TA's instructions.

Exercise 2: Write the following program in a file named `firstprog.cpp`.

```
// This is the first program that just writes out a simple message
// Place your name here
#include <iostream> // needed to perform C++ I/O
using namespace std;
int main ()
{
    cout << "Now is the time for all good men" << endl;
    cout << "To come to the aid of their party" << endl;
    return 0;
}
```

Exercise 3: Compile the program.

Exercise 4: Run the program and write what is printed on the screen.

3 Lab Exercise 2 - Compiling a Program with a Syntax Error

Exercise 1: Write the following program in a file named `semiprob.cpp`.

```
// This program demonstrates a compile error.
#include <iostream>
using namespace std;
int main()
{
    int number;
    float total;
    cout << "Today is a great day for Lab"
    cout << endl << "Let's start off by typing a number of your choice" << endl;
    cin >> number;
    total = number * 2;
    cout << total << " is twice the number you typed" << endl;
    return 0;
}
```

Exercise 2: Compile the program. Here we have our first example of the many syntax errors that you no doubt will encounter in this course. The error message you receive may be different depending on the system you are using, but the compiler insists that a semicolon is missing somewhere. Unfortunately, where the message indicates that the problem exists, and where the problem actually occurs may be two different places. To correct the problem, place a semicolon after the line `cout << "Today is a great day for Lab"`. Most syntax errors are not as easy to spot and correct as this one.

Exercise 3: Re-compile the program and when you have no syntax errors, run the program and input 9 when asked. Record the output.

Exercise 4: Try running it with different numbers. Record your output. Do you feel you are getting valid output?

4 Lab Exercise 3 - Running a Program with a Run Time Error

Exercise 1: Write the following code to a file named `runprob.cpp`.

```
// This program will take a number and divide it by 2.
#include <iostream>
using namespace std;
int main()
{
    float number;
    int divider;
    divider = 0;
    cout << "Hi there" << endl;
    cout << "Please input a number and then hit return" << endl;
    cin >> number;
    number = number / divider;
    cout << "Half of your number is " << number << endl;
    return 0;
}
```

Exercise 2: Compile the program. You should get no syntax errors.

Exercise 3: Run the program. You should now see the first of several run time errors. There was no syntax or grammatical error in the program; however, just like commanding someone to break a law of nature, the program is asking the computer to break a law of math by dividing by zero. It cannot be done. On some installations, you may see this as output that looks very strange. Correct this program by having the code divide by 2 instead of 0.

Exercise 4: Re-compile and run the program. Type 9 when asked for input. Record what is printed.

Exercise 5: Run the program using different values. Record the output. Do you feel that you are getting valid output?

5 Lab Exercise 4 - Working with Logic Errors

Exercise 1: Write the following code to a file named `logicprob.cpp`.

```
// This program takes two values from the user and then swaps them
// before printing the values. The user will be prompted to enter
// both numbers.
#include <iostream>
using namespace std;
int main()
{
    float firstNumber;
    float secondNumber;
    // Prompt user to enter the first number.
    cout << "Enter the first number" << endl;
    cout << "Then hit enter" << endl;
    cin >> firstNumber;
    // Prompt user to enter the second number.
    cout << "Enter the second number" << endl;
    cout << "Then hit enter" << endl;
    cin >> secondNumber;
    // Echo print the input.
    cout << endl << "You input the numbers as " << firstNumber << " and "
         << secondNumber << endl;
    // Now we will swap the values.
    firstNumber = secondNumber;
    secondNumber = firstNumber;
    // Output the values.
    cout << "After swapping, the values of the two numbers are " << firstNumber
         << " and " << secondNumber << endl;
    return 0;
}
```

Exercise 2: Compile the program. You should get no syntax errors.

Exercise 3: Run the program. What is printed?

Exercise 4: This program has no syntax or run time errors, but it certainly has a logic error. This logic error may not be easy to find. Most logic errors create a challenge for the programmer. Find and correct the problem.

Exercise 4: Re-compile and run the program.

Exercise 5: Run the program using different values. Record the output. Do you feel that you are getting valid output?

6 Lab Exercise 5 - Writing Your First Program

Exercise 1: Write a C++ program that will read in a number that represents the number of kilometers traveled. The output will convert this number to miles. 1 kilometer = 0.621 miles. Call this program `kilotomiles.cpp`.

Exercise 2: Compile the program. If you get compile errors, try to fix them and re-compile until your program is free of syntax errors.

Exercise 3: Run the program. Is your output what you expect from the input you gave? If not, try to find and correct the logic error and run the program again. Continue this process until you have a program that produces the correct result.