National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

CS212

Object Oriented Programming

Lab 1: Fundamentals of C++

Class: BEE - 12C

Date: September 20th, 2021

Time: Monday (1400 – 1700)

Name: Muhammad Umer

CMS ID: 345834

Introduction

This Lab will give an overview of the C++ language and how it hangs together, and by introducing a few of the working concepts. Most of this

is just to set the scene before you get into the specifics of writing C++ programs. You'll see what a simple C++ program looks like, and then you'll pull it to pieces to get a rough idea of what the various bits do. You'll also look at the broad concepts of programming in C++ and how you create an executable program from the source code files you'll be writing.

Objectives

The objective of this lab is review of

- Get familiar with C++ environment
- Write a simple C++ program

Tools/Software Requirement

Microsoft Visual Studio 2010/2012/2013

Objective

The main purpose of this lab is to revise the most basic concepts of C++ using Visual Studio.

Description

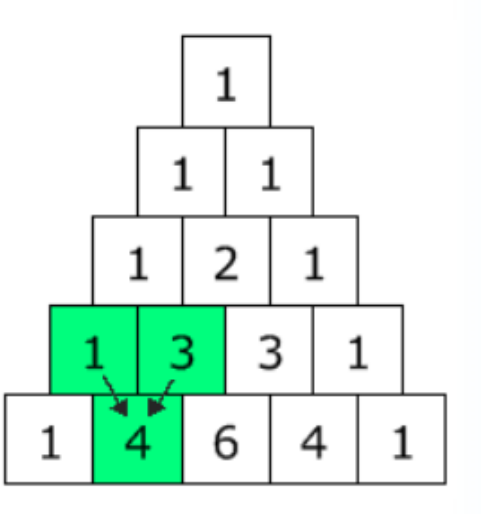
- C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming.
- C++ is regarded as a middle-level language, as it comprises a combination of both high-level and low-level language features.
- C++ was developed by Bjarne Stroustrup starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983.
- C++ is a superset of C, and that virtually any legal C program is a legal C++ program.

Use of C++

- C++ is used by hundreds of thousands of programmers in essentially every application domain.
- C++ is being highly used to write device drivers and other software that rely on direct manipulation of hardware under Realtime constraints.
- C++ is widely used for teaching and research because it is clean enough for successful teaching of basic concepts.
- Anyone who has used either an Apple Macintosh or a PC running Windows has indirectly used C++ because the primary user interfaces of these systems are written in C++.

Lab Tasks

Pascal's triangle is a triangular array of numbers in which those at the ends of the rows are 1 and each of the others is the sum of the nearest two numbers in the row above (the apex, 1, being at the top).



To build the triangle, start with "1" at the top, then continue placing numbers below it in a triangular pattern.
Each number is the numbers directly above it added together.

(Here I have highlighted that 1+3 = 4)

Pascal's Triangle

```cpp
#include <iostream>

using namespace std;

int main() {
    int rows, coef = 1;

    cout << "Enter number of rows: ";
    cin >> rows;

    for (int i = 0; i < rows; i++) {
        for (int space = 1; space <= rows - i; space++)
            cout << "  ";

        for (int j = 0; j <= i; j++) {
            if (j == 0 || i == 0)
                coef = 1;
            else
                coef = coef * (i - j + 1) / j;

            cout << coef << "   ";
        }
        cout << endl;
    }

    return 0;
}
```

Terminal Output

```
D:\NUST\Semester 3\Object Oriented Programming\Labs\Lab 1>lab1.exe
Enter number of rows: 8

                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5   10   10   5   1
    1   6   15   20   15   6   1
  1   7   21   35   35   21   7   1
```

Activity

- **Run the following programs and attach screen shot of output**

A:

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

int main(void) {
    int x = 80;
    int &y = x;
    x++;
    cout << x << " " << --y;
    cin.get();
    return 0;
}
```

Terminal Output

```
D:\NUST\Semester 3\Object Oriented Programming\Labs\Lab 1>lab1.exe
81 80
```

B:

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

long GetNumber(long int Number) {
    return --Number;
}

float GetNumber(int Number) {
    return ++Number;
}

int main(void) {
    int x = 20;
    int y = 30;
    cout << GetNumber(x) << " ";
    cout << GetNumber(y);

    cin.get();
    return 0;
}
```

Terminal Output

```
D:\NUST\Semester 3\Object Oriented Programming\Labs\Lab 1>lab1.exe
21 31
```

Characteristics Of 'C++' Language

1. Modularity

Ability to breakdown a large module into manageable sub modules called as modularity that is an important feature of structured programming languages.

2. Portability

The ability to port i.e., to install the software in different platform is called portability. Highest degree of portability: 'C++' language offers highest degree of portability i.e., percentage of changes to be made to the sources code is at minimum when the software is to be loaded in another platform. Percentage of changes to the source code is minimum.

3. Extendibility

Ability to extend the existing software by adding new features is called as extendibility.

4. Speed

'C++' is also called as middle level language because programs written in 'c++' language run at the speeds matching to that of the same programs written in assembly language so 'c++' language has both the merits of high level and middle level language and because if this feature it is mainly used in developing system software.

Activity

- **Run the following programs and attach screen shot of output**

```
  \/
 \\//
\\\///
///\\\
 //\\
  /\
```

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

int main(void) {
    cout << "  \\/" << endl;
    cout << " \\\\//" << endl;
    cout << "\\\\\\///" << endl;
    cout << "///\\\\\\ " << endl;
    cout << " //\\\\ " << endl;
    cout << "  /\\ " << endl;

    cin.get();
    return 0;
}
```

```
                              Terminal Output
D:\NUST\Semester 3\Object Oriented Programming\Labs\Lab 1>lab1.exe
  \/
 \\//
\\\///
///\\\
 //\\
  /\
```

Lab Tasks

Task 1:

Write a C++ program to print the area and perimeter of a circle.

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

int main() {
    float radius;
    float area, perimeter;

    cout << "Input your circle's radius: " << endl;
    cin >> radius;

    area = (3.14) * (radius * radius);
    perimeter = 2 * 3.14 * radius;

    cout << "Area: " << area << endl;
    cout << "Perimeter: " << perimeter << endl;

    return 0;
}
```

```
                              Terminal Output
D:\NUST\Semester 3\Object Oriented Programming\Labs\Lab 1>lab1.exe
Input your circle's radius:
17
Area: 907.46
Perimeter: 106.76
```

Task 2:

Write down a simple program that takes the length in Feets and inches and then converts it into centimeters.

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

int main(void) {
    int feet;
    float inch, total_inches, centimeters;

    cout << "Input length in feet and inches: " << endl;
    cout << "Example:" << endl << "(feet)" << endl << "(inches)" << endl;
    cin >> feet >> inch;

    total_inches = (feet * 12) + inch;
    centimeters = total_inches * 2.54;
    cout << "Input in Centimeters is: " << centimeters;

    return 0;
}
```

Terminal Output

```
D:\NUST\Semester 3\Object Oriented Programming\Labs\Lab 1>lab1.exe
Input length in feet and inches:
Example:
(feet)
(inches)
5
9.5
Length in Centimeters: 176.53
```

Task 3:

Write down a simple program which can convert a String into all Upper Case?

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

int main(void) {
    string inputString;

    // Taking the whole line as input
    cout << "Input a string: " << endl;
    getline(cin, inputString);

    // Converting all lowercase alphabets to their upper counterparts
    for (int i = 0; i < inputString.length(); i++) {
        if(inputString[i] >= 97 && inputString[i] <= 122 ) {
            inputString[i] = inputString[i] - 32;
```

```
        }
    }
    cout << "Uppercased String: " << inputString;

    return 0;
}
```

Terminal Output

```
D:\NUST\Semester 3\Object Oriented Programming\Labs\Lab 1>lab1.exe
Input a string:
Hello World!
Uppercased String: HELLO WORLD!
```

Task 4:

Write a Program that can take five real numbers and then able to sort them and display them in the Descending Order.

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

int main ()
{
    int numbers[5], i, j, temp;

    // Taking the inputs
    cout << "Enter 5 Numbers: " << endl;

    for (i = 0; i < 5; ++i) {
        cin >> numbers[i];
    }
    // Main Algorithm
    for (i = 0; i < 5; ++i)
    {
        for (j = i + 1; j < 5; ++j)
        {
            if (numbers[i] < numbers[j])
            {
                temp = numbers[i];
                numbers[i] = numbers[j];
                numbers[j] = temp;
            }
        }
    }
```

```
    // Output
    cout << "Descending Order: " << endl;
    for (i = 0; i < 5; ++i)
    {
        cout << numbers[i];
        cout << " ";
    }

    return 0;
}
```

Terminal Output

```
D:\NUST\Semester 3\Object Oriented Programming\Labs\Lab 1>lab1.exe
Enter 5 Numbers:
123 45 23 87 66
Descending Order:
123 87 66 45 23
```
