Lab 01

C++ Revision and Array List

Objective:

In this lab, students will get familiar with the new language C++ and its IDE with the help of simple programs that will clear the syntax of the language. Beside learning the basics of C++ like control statements, functions, arrays, etc., you will also learn the concept of Array List.

Activity Outcomes:

This lab teaches you the following topics:

- Basic syntax of C++
- Data types and operators in C++
- Control flow statements in C++
- Arrays and Functions

Instructor Note:

As a pre-lab activity, read fundamental concepts from the text book "C++ How to Program, Deitel, P. & Deitel, H., Prentice Hall, 2019".

1) Useful Concepts

C++, as we all know is an extension to C language and was developed by **Bjarne stroustrup** at bell labs. C++ is an intermediate level language, as it comprises a confirmation of both high level and low level language features.
Following features of C++ makes it a stronger language than C,

1. There is Stronger Type Checking in C++.
2. C++ supports and allows user defined operators (i.e Operator Overloading) and function overloading is also supported in it.
3. Exception Handling is there in C++.
4. Inline Functions in C++ instead of Macros in C language. Inline functions make complete function body act like Macro, safely.
5. Variables can be declared anywhere in the program in C++, but must be declared before they are used.

Installing GNU C/C++ Compiler

Install GCC at Windows you need to install MinGW. To install MinGW, go to the MinGW homepage, www.mingw.org, and follow the link to the MinGW download page. Download the latest version of the MinGW installation program which should be named MinGW-<version>.exe.

While installing MinGW, at a minimum, you must install gcc-core, gcc-g++, binutils, and the MinGW runtime, but you may wish to install more.

Add the bin subdirectory of your MinGW installation to your **PATH** environment variable so that you can specify these tools on the command line by their simple names.

When the installation is complete, you will be able to run gcc, g++, ar, ranlib, dlltool, and several other GNU tools from the Windows command line.

IDE for coding

The IDE we will be using is Dev-C++. Today we will clear our concepts regarding the syntax of C++ with the help of small programs.

Note: we use cin for input and cout for output/print.

At start of your program always use

```
#include<iostream>

using namespace std;
```

Shortcut Keys:

- Press F10 to compile.
- Press F9 to run the complied program.
- Press F11 to compile and run the program.
- For functions in C++ you will have to define them before writing the main method and the main methods always return 0;

2) Solved Lab Activites

Sr.No	***Allocated Time***	***Level of Complexity***	***CLO Mapping***
Activity 1	***5 mins***	***Low***	***CLO-4***
Activity 2	***5 mins***	***Low***	***CLO-4***
Activity 3	***5 mins***	***Low***	***CLO-4***
Activity 4	***5 mins***	***Low***	***CLO-4***
Activity 5	***5 mins***	***Low***	***CLO-4***
Activity 6	***5 mins***	***Low***	***CLO-4***
Activity 7	***5 mins***	***Low***	***CLO-4***
Activity 8	***5 mins***	***Low***	***CLO-4***
Activity 9	***5 mins***	***Low***	***CLO-4***
Activity 10	***5 mins***	***Low***	***CLO-4***
Activity 11	***5 mins***	***Low***	***CLO-4***
Activity 12	***5 mins***	***Low***	***CLO-4***

Activity 1:

Single Line, Multi Line comments and basic Input/ Output in C++.

Solution:

```
// Single line Comment

/*
```

```
* Multiple line
* comment
*/
#include<iostream>
using namespace std;
int main()
{
cout<<"Hello World!";
return 0;
}
```

Output

The output will be “Hello World!”.

Activity 2:

Variables are containers for storing data values.
In C++, there are different types of variables (defined with different keywords), for example:

- **int** - stores integers (whole numbers), without decimals, such as 123 or -123
- **double** - stores floating point numbers, with decimals, such as 19.99 or -19.99
- **char** - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
- **string** - stores text, such as "Hello World". String values are surrounded by double quotes
- **bool** - stores values with two states: true or false

Solution:

```
#include <iostream>
using namespace std;
int main ()
{
  float a=5.5;               // initial value: 5
  int b(3);              // initial value: 3
  int c{2};              // initial value: 2
  float result;            // initial value undetermined

  a = a + b;
  result = a - c;
```

```
   cout << result;


   return 0;

 }
```

Output

Value present in result variable will be shown on Screen

Activity 3:

Using cin, extraction operator (>>) and cout (<<). cin iJava program to illustrate enhanced for loop

```
#include <iostream>

using namespace std;

int main ()

{

int x;
cout << "Type a number: "; // Type a number and press enter
cin >> x; // Get user input from the keyboard
cout << "Your number is: " << x; // Display the input value

return 0;

}
```

Output

Type a number: 100   //suppose user enters 100

Your number is: 100

Activity 4:

Write a program to use string data type in C++

```
#include <iostream>

#include <string>

using namespace std;

int main ()
```

```
{
  string mystring;
  mystring = "This is the initial string content";
  cout << mystring << endl;
  mystring = "This is a different string content";
  cout << mystring << endl;
  return 0;
}
```

Output

```
This is the initial string content
This is a different string content
```

Activity 5:

Write a program to use arithmetic operators in C++

```
#include <iostream>
using namespace std;
int main ()
{
  int a, b=3;
  a = b;
  a+=2;             // equivalent to a=a+2
  cout << a;
}
```

Output

```
5
```

Activity 6:

Write a program to use relational operators in C++

```
#include <iostream>
using namespace std;
int main ()
{
  int a,b,c;

  a=2;
  b=7;
  c = (a>b) ? a : b;

  cout << c << '\n';
}
```

Output

7

Activity 7:

Write a program to use if-else statement in C++

```
#include <iostream>
using namespace std;
int main ()
{
int x;
cin>>x;
if (x > 0)
  cout << "x is positive";
else if (x < 0)
  cout << "x is negative";
else
```

```
  cout << "x is 0";
}
```

Output

The ouput depends upon the value of ‘x’ entered by the user, the output will be either x is positive or x is negative or x is 0.

Activity 8:

Write a program to use while loop in C++

```
#include <iostream>
using namespace std;
int main ()
{
  int n = 10;

  while (n>0) {
    cout << n << ", ";
    --n;
  }

  cout << "liftoff!\n";
}
```

Output

10, 9, 8, 7, 6, 5, 4, 3, 2, 1, liftoff!

Activity 9:

Write a program to use do-while loop in C++

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

int main ()
{
  string str;
  do {
    cout << "Enter text: ";
    getline (cin,str);
    cout << "You entered: " << str << '\n';
  } while (str != "goodbye");
}
```

Output

Program repeatedly takes a string as input then displays it saying "You entered…." Until user types goodbye.

Activity 10:

Write a program to apply the switch statements.

```
#include <iostream>
using namespace std;
int main ()
{
int day = 4;
switch (day) {
  case 1:
    cout << "Monday";
    break;
  case 2:
    cout << "Tuesday";
    break;
  case 3:
    cout << "Wednesday";
    break;
  case 4:
    cout << "Thursday";
    break;
  case 5:
    cout << "Friday";
    break;
  case 6:
    cout << "Saturday";
    break;
  case 7:
    cout << "Sunday";
    break;

default:

cout<<"Enter valid day";
}


}
```

Output

Outputs "Thursday" as variable day is initialized to 4

Activity 11:

Write a program that will add two numbers in a function and call that function in main.

```
#include <iostream>

using namespace std;

int addition (int a, int b)

{

  int r;

  r=a+b;

  return r;

}


int main ()

{

  int z;

  z = addition (5,3);

  cout << "The result is " << z;

}
```

Output

The result is 8

Activity 12:

Write a program to use the concept of Arrays in C++

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

int foo [] = {16, 2, 77, 40, 12071};
int n, result=0;

int main ()
{
  for ( n=0 ; n<5 ; ++n )
  {
    result += foo[n];
  }
  cout << result;
  return 0;
}
```

Output

12206

3) Graded Lab Tasks

Note: The instructor can design graded lab activities according to the level of difficulty and complexity of the solved lab activities. The lab tasks assigned by the instructor should be evaluated in the same lab

Lab Task 1

Write down a program that find $\sum X^2$ *, where input for X and starting and stopping value is entered by the user.*

Lab Task 2

Write the computer program to apply the concepts of Array List. The array list will include the following functions:

1. *Insert the value at end of the list*
2. *Insert the value at start of the list*
3. *Insert the value after specific value*
4. *Insert the value before specific value*
5. *Display the array list*
6. *Delete the value from end of the list*
7. *Delete the value from start of the list*
8. *Delete specific value*

Lab Task 3

Using while loop apply liear search on the Array List you developed in Lab Task 2.