Introduction to I DE & Introduction to C+ +

LAB # 1

2020

CSE102L Computer Programming Lab

Submitted to:

Engr. Abdullah Hamid

Submitted by:

TAYYABA

Registration No **:**

19PWCSE1854

Semester: **2nd**

Class Section: **C**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _____

March 2nd, 2020

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Introduction to IDE and Introduction to C++

Objectives:-

1. To be able to install and use Code::Blocks IDE for compiling C++ programs
2. To be familiar with syntax and structure of C++ programming

IDE:-

IDE stands for **I**ntegrated **D**evelopment **E**nvironment. It is a software application that provides comprehensive facilities to computer programmers for software development. An **IDE** normally consists of at least a source code editor, build automation tools and a debugger.

History of C++:-

Many new programming languages appeared during the 1960s. The computers at that time were in early stage of development. The language **ALGOL 60** was an alternative to **FORTAN.** The language **CPL** (**C**ombined **P**rogramming **L**anguage) was developed in 1963. It was more specific for concrete programming task of that time than **ALGOL 60** and **FORTAN.**

Ken Thompson created **B** language in 1970. **C** language was derived from B. the B language provided the basis of development of **C. C** language was designed to write system program under **UNIX** operating system. The power and flexibility made C popular in industry for wide range of application.

The earlier version of C was known as **K&R** (**K**ermighan **and R**itche) **C.** The **A**merican **N**ational Standard **I**nstitute (**ANSI**) developed the standard version of the language. The standard version of **C** known as **ANSI C.**

Bjarne Stroustrup from **Bell Labs** started the development of C++ language in 1980. It was first commercial release of the language appeared in October 1985. It was originally named "C with Classes". C++ was refined in 1980s and it was a unique language. It was very much compatible with the code of C and provided the most important characteristics of C.**ANSI** committee **X3J16** begun the development of specific standard for C++ from 1990. In mid 1998 **ANSI/ISO C++** language standards were approved. This standard was used by most of compilers used today to compile program. The language became very popular and now it is preferred to develop professional application on all platforms.

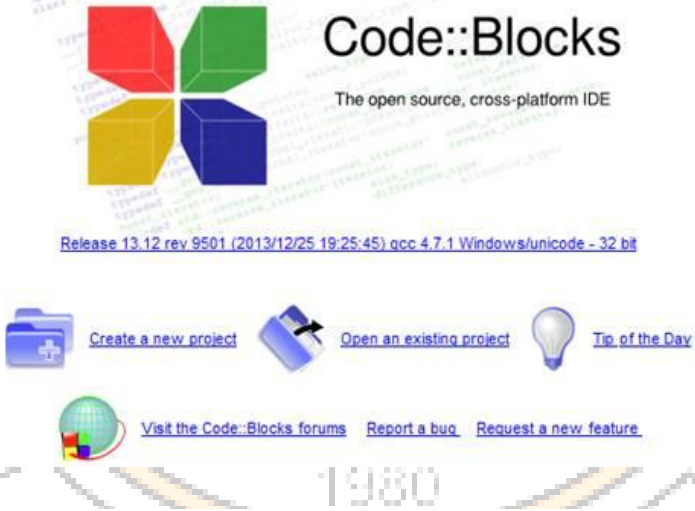
Getting use to with Code blocks:-

Downloading and installing: Code Blocks can be downloaded freely from its official website, www.code::block.org. Identifying the correct package is the first essential task, because there are couple distinct packages available, leveraging dispersed features for both Windows and Linux platforms.

Windows 2000 / XP / Vista / 7:

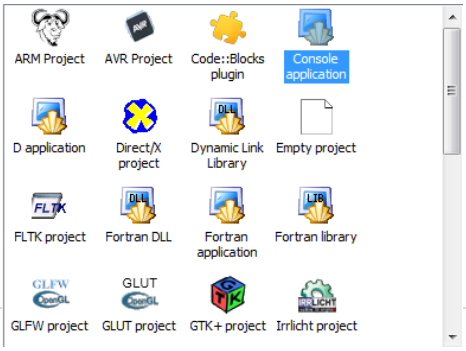
File	Date	Download from
codeblocks-13.12-setup.exe	27 Dec 2013	BerliOS or Sourceforge.net
codeblocks-13.12mingw-setup.exe	27 Dec 2013	BerliOS or Sourceforge.net
codeblocks-13.12mingw-setup-TDM-GCC-481.exe	27 Dec 2013	BerliOS or Sourceforge.net

Once you have downloaded the correct package, its installation is quite easy on Windows. It installs like any other typical software. Finally, the Code::Blocks development environments startup window.



Entering to interface and coding: After you are done with installation and subsequent configuration, it's time to start coding. You will observe a screen appears right after imitating this software that enables you to create a new project and other functionalities.

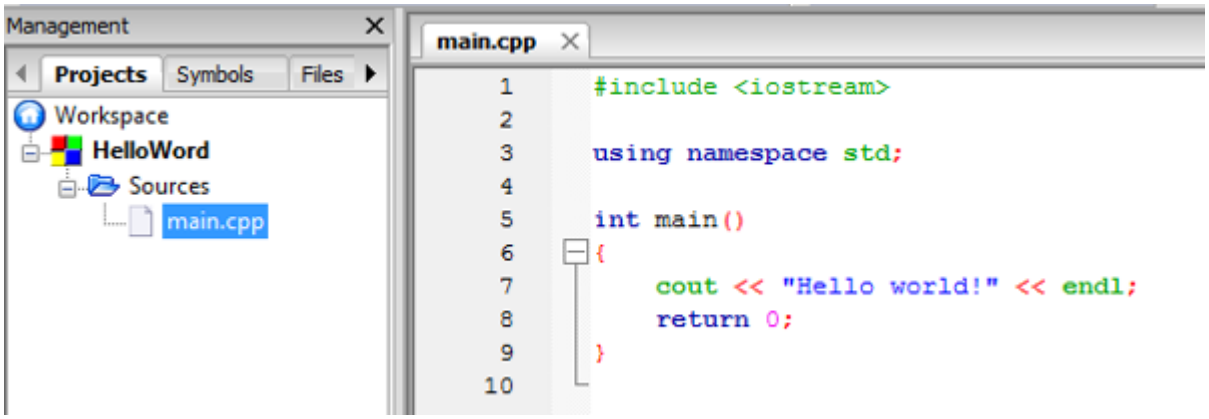
To start a new project, click 'Create New Project' on the screen. Here, you will encounter with a huge list of predefined project templates. Go ahead and select "Console Application;" this will allow you to write a program for the console.



The other application templates in above figure are for developing more advanced types of applications. After

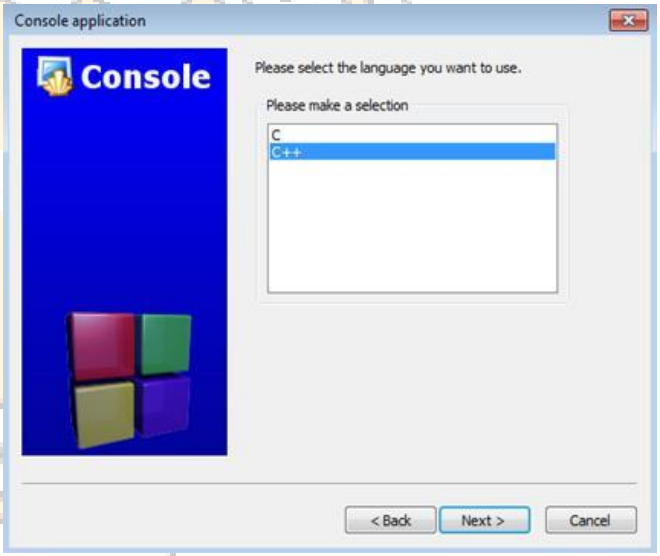
selecting Console Application, click the Go button to begin using the Console Application Wizard. Henceforth, the wizard will ask to choose the programming language between C and C++ for coding like as mentioned in figure below.

Name your file. Choose the name you might remember after opening long time opening, and start it. Type a program of C++ as given below:



You will get output “Hello World”.

Preprocessor Directives:-



Preprocessor directive are instruction given to compiler before execution of program. Preprocessor directive are also called **compiler directive**. Preprocessor directive start with # (Hash) symbol. There is most common preprocessor directive is include preprocessor.

Include Preprocessor:-

Include preprocessor is used to include header file in the program. The syntax for include preprocessor is:-

#include <header_file> or #include ”header_file”.

For example:-

#include<iostream>

Header File:-

Header file is the collection of **library functions** to perform different task. There are different header files used for different purposes. Each header file contains different types predefined functions. The include preprocessor is used to call header file.

Main () Function:-

Main () function is the starting point of C++ program. When the program runs, the control enters to the main () function and start executing its statements. A program must contain a main function. If there is no main () function in the program, the program will be compiled but not executed. The **syntax** for main function is:

```
Main ()
{
    Body of program.
}
```

Body of a Program:-

The body of program contains different statement to be performed. These statements are written in curly brackets. These curly brackets are called delimiters. Each statement should end with a statement terminator (;) in the end.

For example:

```
#include<iostream>
using namespace std;
int main()
 {
        cout<<"Hello World";
        return 0;
}
```

Variable:-

A **variable** is a name given to memory location. It is basic unit of storage in a program. The value in variable can be changed during program execution. A variable is only name given to memory location, all the operation done on variable effects that memory location.

There two types of Variable:

Global variable: A global variable is a variable whose value extinct for more than one functions. There values can be used in more than one function.

Local variable: A local variable is a variable whose value extinct for only one function. It means that value of a variable can be used for only one function.

Identifiers:

Identifiers are the names used to represent **variables, constants, types, functions** and **labels** in the program. Identifier is important feature for all computer languages. A good identifier should be descriptive and short. There are some important rules for identifier name are following:-

1. The first character must be alphabetic or underscore (_).
2. The identifier name must consist of alphabetic characters, numbers and underscore (_).
3. The reserved words cannot be used as identifier name.

There are two types of identifiers:

Standard identifier: A type of identifier that has special meaning to C++ is known as **standard identifier**. C++ cannot use standard identifier for its original purpose if it is redefined. **cin** and **cout** are examples of standard identifier.

User-defined identifier: The type of identifier that is defined by the programmer to across memory location is known as **user-defined identifier**. User-defined identifiers are used to store **data** and **program results**. **Age** and **Result** are examples of user-defined identifiers.

Data Types:-

The data types define a set of values and a set of operations on those values. The computer manipulates various types of data type. The data is given to a program as input. The data is processed

according to the program instruction and output is returned. The type of each data value is identified at the beginning of program design. A C++ may need to process different types of data. Each data type requires a different amount of memory. C++ provides following data types:

Data Type	Purpose
int	To store numeric values
float	To store real(decimal points) values
Double	To store large real(decimal points) values
Char	To store character values

Integer Data Type:-

Integer data type includes data which have numeric value with no decimal point or fraction. It includes both positive and negative sign numbers. There are different integer data types:-

Data Type	Size in Bytes	Description
int	2	Ranges from -32,768 to 32,767.
Short	2	Ranges from -32,768 to 32,767.
unsigned int	2	Ranges from 0 to 65,535.
Long	4	Ranges from -2,147,483,648 to 2,147,483,647.
Unsigned long	4	Ranges from 0 to 4,294,967,295.

Real Data Types:-

Real data is numeric value with decimal point or fraction. It is also known as floating point numbers. It includes both positive and negative numbers. There are different real data types:-

Data Type	Size in Bytes	Description
float	4	$3.4*10^{-38}$ to $3.4*10^{+38}$
double	8	$1.7*10^{-308}$ to $1.7*10^{+308}$
long double	10	$1.7*10^{-4932}$ to $1.7*10^{+4932}$

Character Data Type:-

Char data type is used to store character value. It takes **1** byte in memory. It is used to represent a **letter, number** or **punctuation mark** or few other **symbols**.

Programming task

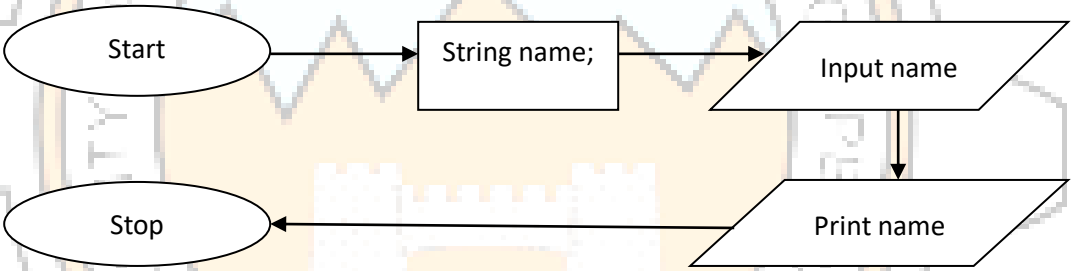
1. **Write a program to display your name on console.**

Code:

[*] 1st_Lab_rep.cpp

```
#include<iostream>
using namespace std;
main()
{

    string name;
    cout<<"Please Enter Your Name \n";
    cin>>name;
    cout<<"Assalam o Alikum  "<<name<<" \n Welcome to Dev_C++ .";

}
```

Flow Chart:



Output:

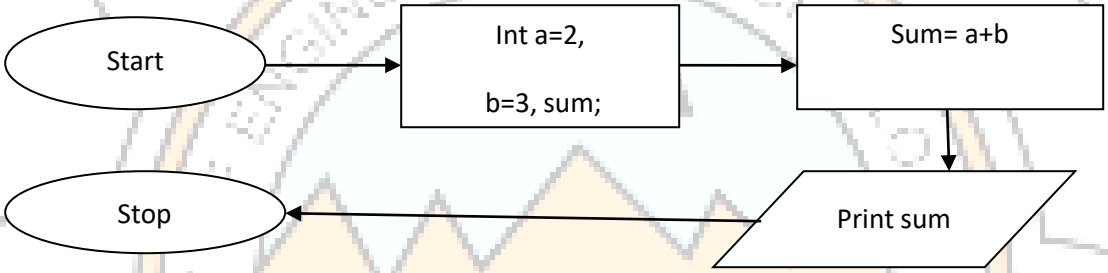
```
C:\Users\Pc\Documents\CPP Lab pactice\1st_Lab_rep.exe
Please Enter Your Name
Tayyaba
Assalam o Alikum  Tayyaba
 Welcome to Dev_C++ .
--------------------------------
Process exited after 9.143 seconds with return value 0
Press any key to continue . . .
```

2. **Write a program to add two numbers (2+3=) and display its sum.**

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int a=2,b=3,sum;
    sum=a+b;
    cout<<"The Sum of Two Numbers "<<a<<" & "<<b<<" is equal to "<<sum;
}
```

Flow chart:-



Output:

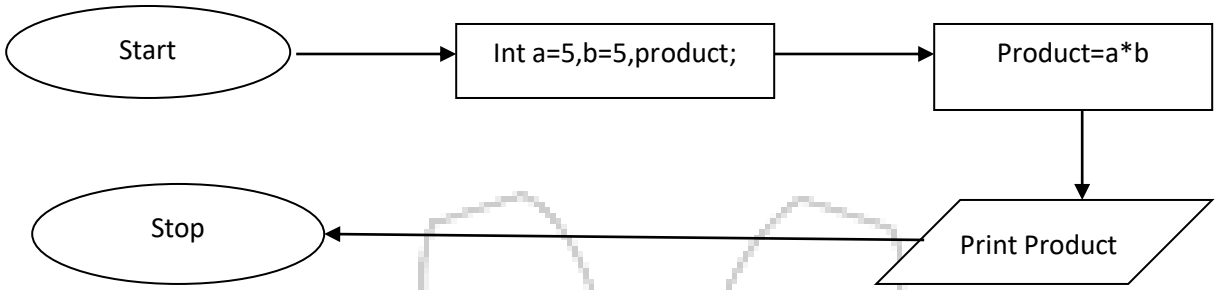
```
The Sum of Two Numbers 2 & 3 is equal to 5
--------------------------------
Process exited after 1.406 seconds with return value 0
Press any key to continue . . .
```

3. Write a program to multiply two numbers (5x5=) and display its product.

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int a=5,b=5,product;
    product=a*b;
    cout<<"The Product of Two Numbers "<<a<<" & "<<b<<" is equal to "<<product;
}
```

Flow Chart:-



Output:

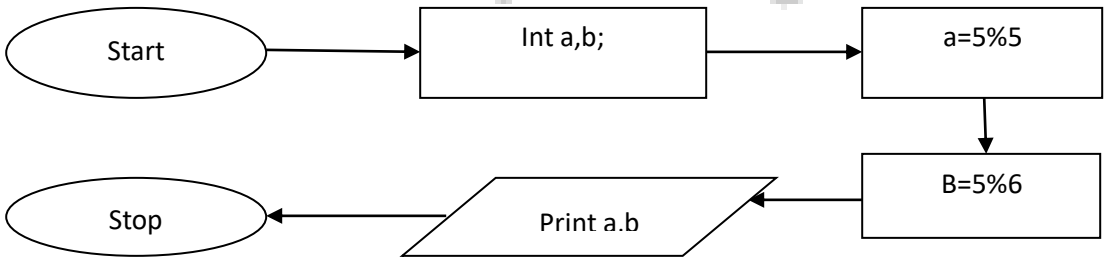
```
The Product of Two Numbers 5 & 5 is equal to 25
--------------------------------
Process exited after 0.9918 seconds with return value 0
Press any key to continue . . .
```

4. **Write a program to find the mod of (5%5=) and (5%6=)**

Code:

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    a=5%5;
    b=5%6;
    cout<<" 5 % 5 = "<<a<<endl;
    cout<<" 5 % 6 = "<<b<<endl;
}
```

Flow Chart:-



Output:

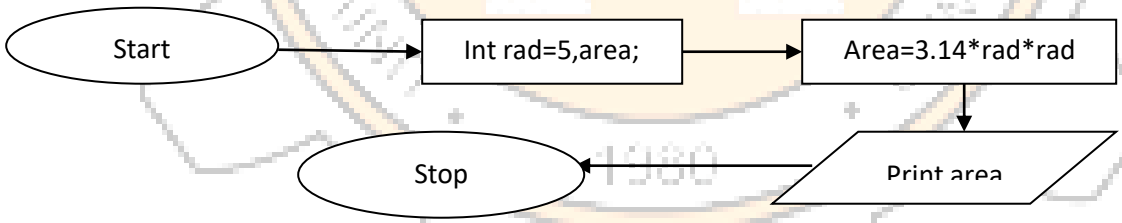
```
 5 % 5 = 0
 5 % 6 = 5

--------------------------------
Process exited after 0.05734 seconds with return value 0
Press any key to continue . . .
```

5. **Write a program to calculate area of a circle having its radius (r = 5).**

Code:-

```
#include<iostream>
using namespace std;
int main()
{
    int rad=5,area;
    area=3.14*rad*rad;
    cout<<"The area of circle whose radius is "<<rad<<" is equal to "<<area;
}
```

Flow Chart:-



Output:

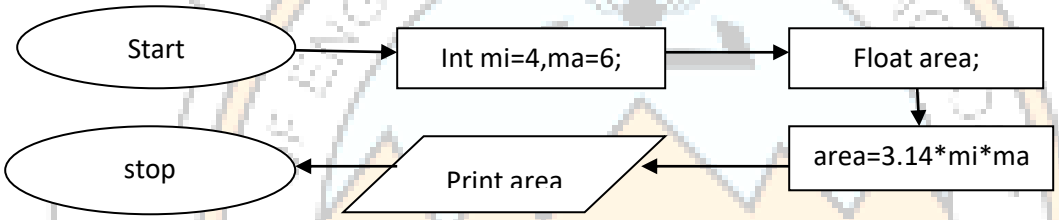
```
The area of circle whose radius is 5 is equal to 78
--------------------------------
Process exited after 0.9692 seconds with return value 0
Press any key to continue . . .
```

6. **Write a program to calculate area of an ellipse having its axes (minor=4cm, major=6cm).**

Coding:

```
#include<iostream>
using namespace std;
int main()
{
    int mi=4,ma=6;
    float area;
    area=3.14*mi*ma;
    cout<<"The area of ellipse is eqeal to "<<area;
}
```

Flow chart:-



Output:

```
The area of ellipse is eqeal to 75.36
--------------------------------
Process exited after 1.376 seconds with return value 0
Press any key to continue . . .
```