Behind the code

```
#include<iostream>
using namespace std;
int main()
{
    cout<<”Welcome in InnoLabZ”
}
```

I hope everyone encounters these lines of code in their starting of engineering. Many students mug up these lines of code without knowing the reason behind it. So, in this blog I will try to elaborate the reason behind each symbol and lines and try to ease the process of programming understanding. Let's start…

There is a basic structure of any code. The source code written by programmers is stored in the file using the .cpp extension. This file is then processed by preprocessors and an expanded source code file is generated named a program like sum.cpp. This expanded file is compiled by compiler and an object code file is generated named program .obj like sum.obj. Finally, the linker links this object code file to the object code of the library functions to generate the executable file program.exe like sum.exe.

Here we explain the code of summation of 2 numbers. Below is the source code of summation of two numbers.

```
#include<iostream>              // Header file (1)
using namespace std;
int main()                      // Main function  (2)
{
    int a,b,c;                  //variable declaration (3)
    cin>>a>>b;                  // input from user  (4)
    c = a+b;                    //summation operation  (5)
    cout<<”Sum = ”<<c;          // Print output   (6)
    return 0;                    //exit (7)
}
```

(//) this represents the comment in C++ programming language. Comment is used to make code more readable to every person.

(;) semicolons represent the end of any statement.

1) **Header file**

#include

The ‘#’ symbol indicates whatever statement starts with #, is going to the preprocessor of the program, and the preprocess program will execute this statement. Examples of some preprocessor directives are: *#include*, *#define*, *#ifndef* etc. Remember # symbol only provides a path that it will go to the preprocessor, and commands such as **include** are processed by the preprocessor program. For example, **include** will include extra code to your program.

Preprocessor programs provide preprocessors directives which tell the compiler to preprocess the source code before compiling. All preprocessor directives begin with a ‘#’ (hash) symbol.

<iostream>

It is a file of all input/output standards of c++. It contains many files that are used in code like cin,cout files.

Using namespace std

“using namespace std” means we use the namespace named std. “std” is an abbreviation for standard. So that means we use all the things within the “std” namespace. If we don’t want to use the (using namespace std) line of code, we can use namespace like this. std::cout, std::endl. If this namespace is not used the computer finds for the cout, cin, endl and computer cannot identify those and therefore it throws errors.

2) **Main function**

int main()

Every code must be a main function. Execution of code starts from the main function. Every main function contains ( ) small brackets,under this argument of function call is written as per requirement. Open curly braces { show start of function and closed curly braces } show end of function.

3) **Variable declaration**

int a;

Here, **int** is a predefined data type and **a variable** name. **int a** defines **a variable** having integer data type.

4) Input

```
cin>>a>>b;
```

Cin is a function of an iostream file. This function used to take the input from the user.cin must be followed by >> this symbol. Eg. let for above code users give input a=5 & b=6.

5) Summation operation

```
c=a+b;
```

This statement performs the addition task on two variables a & b and stores the result into c variable. Eg. we take input a=5 & b=6 and add them to store them into c as 11.

6) Output

```
cout<<”Sum = ”<<c;
```

Cout is the function to print the output of a program in c++. Print the content as it under the quotation mark(“ ”). cout is must followed by << this symbol. Eg. Here o/p is Sum = 11.

7) Exit

```
return 0;
```

The return 0; statement is Exit status of the programm. The program ends with this statement.

I hope everyone enjoyed it after reading and visualizing the process behind the code. Hurray!….