C++ Programming Basics :-

C++ is a general-purpose programming language and widely used nowadays for competitive programming. It has imperative, object-oriented and generic programming features. C++ runs on lots of platform like Windows, Linux, Unix, Mac, etc.

However to become proficient in any programming language, one Firstly needs to understand the basics of that language.

Therefore, below are the basics of C++ in the format in which it will help you the most to get the headstart:

Basic Syntax and First Program in C++: Learning C++ programming can be simplified into writing your program in a text editor and saving it with correct extension(.CPP, .C, .CP) and compiling your program using a compiler or online IDE. The “Hello World” program is the first step towards learning any programming language and also one of the simplest programs you will learn.

Basic I/O in C++:C++ comes with libraries which provides us with many ways for performing input and output. In C++ input and output is performed in the form of a sequence of bytes or more commonly known as streams. The two keywords cin and cout are used very often for

taking inputs and printing outputs respectively. These two are the most basic methods of taking input and output in C++.

Comments in C++: A well-documented program is a good practice as a programmer. It makes a program more readable and error finding become easier. One important part of good documentation is Comments. In computer programming, a comment is a programmer-readable explanation or annotation in the source code of a computer program. These are statements that are not executed by the compiler and interpreter.

Data Types and Modifiers in C++: All variables use data-type during declaration to restrict the type of data to be stored. Therefore, we can say that data types are used to tell the variables the type of data it can store. Whenever a variable is defined in C++, the compiler allocates some memory for that variable based on the data-type with which it is declared. Every data type requires a different amount of memory.

Uninitialized variable in C++: “One of the things that has kept C++ viable is the zero-overhead rule: What you don’t use, you don’t pay for.” -Stroustrup. The overhead of initializing a stack variable is costly as it hampers the speed of execution, therefore these variables can contain indeterminate values. It is considered a best practice to initialize a primitive data type variable before using it in code.

Undefined Behaviour in C++: If a user starts learning in C/C++ environment and is unclear with the concept of undefined behaviour then that can bring plenty of problems in the future like while debugging someone else code might be actually difficult in tracing the root to the undefined error.

Variables and Types in C++: A variable is a name given to a memory location. It is the basic unit of storage in a program. The value stored in a variable can be changed during program execution. A variable is only a name given to a memory location, all the operations done on the variable effects that memory location. In C++, all the variables must be declared before use.

Variable Scope in C++: In general, scope is defined as the extent up to which something can be worked with. In programming also the scope of a variable is defined as the extent of the program code within which the variable can be accessed or declared or worked with. There are mainly two types of variable scopes, Local and Global Variables.

Constants and Literals in C++: As the name suggests the name constants is given to such variables or values in C++ programming language which cannot be modified once they are defined. They are fixed values in a program. There can be any types of constants like integer, float, octal, hexadecimal, character constants, etc. Every constant has some range. The integers that are too big to fit into an int will be taken as long. Now there are various ranges that differ from unsigned to signed bits. Under the signed bit, the range of an int varies from -128 to +127 and under the unsigned bit, int varies from 0 to 255. Literals are kind of constants and both the terms are used interchangeably in C++.

Types of Literals in C++: In this article we will analyse the various kind of literals that C++ provides. The values assigned to each constant variables are referred to as the literals. Generally, both terms, constants and literals are used interchangeably. For eg, “const int = 5;“, is a constant expression and the value 5 is referred to as constant integer literal.

Access Modifiers in C++: Access modifiers are used to implement an important feature of Object-Oriented Programming known as Data Hiding. Access modifiers or Access Specifiers in a class are used to set the accessibility of the class members. That is, it sets some restrictions on the class members not to get directly accessed by the outside functions.

Storage Classes in C++: Storage Classes are used to describe the features of a variable/function. These features basically include the scope, visibility, and life-time which help us to trace the existence of a particular variable during the runtime of a program.

Operators in C++: Operators are the foundation of any programming language. Thus the functionality of C/C++ programming language is incomplete without the use of operators. We can define operators as symbols that help us to perform specific mathematical and logical computations on operands. In other words, we can say that an operator operates the operands.

Loops in C++: Loops in programming comes into use when we need to repeatedly execute a block of statements. For example: Suppose we want to print “Hello World” 10 times. This can be done in two ways, Iterative method and by using Loops.

Decision Making in C++:There comes situations in real life when we need to make some decisions and based on these decisions, we decide what should we do next. Similar situations arise in programming also where we need to make some decisions and based on these decisions we will execute the next block of code. Decision-making statements in

programming languages decide the direction of flow of program execution.

Forward declarations in C++: It refers to the beforehand declaration of the syntax or signature of an identifier, variable, function, class, etc. prior to its usage (done later in the program). In C++, Forward declarations are usually used for Classes. In this, the class is pre-defined before its use so that it can be called and used by other classes that are defined before this.

Errors in C++: Error is an illegal operation performed by the user which results in abnormal working of the program. Programming errors often remain undetected until the program is compiled or executed. Some of the errors inhibit the program from getting compiled or executed. Thus errors should be removed before compiling and executing.