

C++ Programming

Sadik Saifi

Oct 24, 2022

Contents

Basics Of Programming Languages	2
Types Of Programming Languages	2
Low Level Programming Languages	2
High Level Programming Languages	2
C++ Programming Language.	4
C++ Sample Program.	4
C++ Variables	4
C++ Data Types	5
C++ Operators	5
Conditionals Statements	6
If else Statements	6
Nested If else Statements	6
Ternary Statements	6
Switch Case Statements	6
C++ Loops/Iterative Statements	7
Types Of Loops	7
While Loop	7
For Loop	7
For Loop vs While Loop	7
Do-while Loop	7
Function	8
How to declare a function.	8
Function Prototype	8

Basics Of Programming Languages

Programming is the process of creating a set of instructions which tells a computer how to perform a task.

Types Of Programming Languages

- Low Level Languages.
- High Level Languages.

Low Level Programming Languages

Assembly Level Programming Language Low-level programming that is intended to communicate directly with hardware.

High Level Programming Languages

Easy to understand and less complex than assembly level language/machine code.

Types Of High Level Programming Languages.

- Procedural
- Functional
- Object-Oriented

Procedural:

- It is written in set of procedures which executes in a structure/serial order.
- Procedures are also known as subroutine/function.
- e.g. C, Basic, Pascal etc.

Functional:

- We functions to write the code.
- Functions are chunks/block of code which can use over and over again.
- They can also take parameters.
- e.g. Python, JavaScript etc.

Object-Oriented:

- We use object and classes.
- Classed are user defined prototype/blueprints which can be used to create objects.
- Object are real life entity.
- e.g. C++, Java, C# etc.

Class: We can define some properties, attributes, methods etc.

Object: Instance of a class which allows to use variables and methods from class.

CPP Programming Language.

- It is Object-Oriented Programming Language.
- But it also has support for procedural programming.
- Initially it was intended to developed as a superset of C but later on it became a new programming language.
- Developed by **Bjarne Stroustrup** in **1979**.

C++ Sample Program.

```
#include<iostream>    /*iostream is directive & it is processed by
                        preprocessor.
                        Preprocessor is program that compiler runs.
                        #include telling our program to include the header
                        files like iostream.
                        iostream contains input output functions.
                        */
using namespace std;  /*It is telling our program that there is namespace of
                        name std which we have to use in our program.
                        e.g cout belongs to std namespace
                        - std:cout
                        */
int main() {          /*'int main' is the main function which the entry point
                        of a program.*/
    int val;          //declaring a variable.
    cout<<"Hello";    //printing the output.
    cin>>val;         //taking input.
    cout<<val;        //printing the output.
    return 0;         //return 0 indicates, program has executed successfully.
    cout<<"Hi";       //this won't be executed.
}
```

C++ Variables

- Variables are just containers to store our value where our code is executed.
- As C++ is statically typed language, so you can't store different type of value in different type of container.
e.g.

```
int a = 5;           // correct
int b = "f";         // incorrect
int e = "6";         // incorrect
char c = "a"         // correct
char d = "4"         // correct
```

C++ Data Types

Primary	Derived	User Defined
Integer	Function	Class
Character	Array	Structure
Boolean	Pointer	Union
Floating Point	Reference	Enum
Double Floating Point	-	-
Void	-	-
Wide Character		

C++ Operators

- Arithmetic Operators
 - e.g. +, -, *, /, %, ++, - etc.
- Relational Operators
 - e.g. ==, !=, >, <, >=, <= etc.
- Logical Operators
 - e.g. &&, ||, ! etc.
- Assignment Operators
 - e.g. =, +=, -=, /=, %= etc.
- Bitwise Operators
 - e.g. ~, <, >, |, &, ^ etc.
 - $a \ll b = a \times 2^b$
 - $a \gg b = a / 2^b$
- Misc Operators
 - e.g. sizeof, ?exp1:exp2, comma Operator, dot & arrow Operators, casting Operator, & Address Operator, * Pointer Operator etc.
- Unary Operators
 - e.g. +, -, ++, --, ! etc.

Conditionals Statements

- If else Statements.
- Nested If else Statements
- Ternary Statements

If else Statements

```
if (condition) {  
    // block of code if condition is true  
}  
else {  
    // block of code if condition is false  
}
```

Nested If else Statements

```
if (condition1) {  
    // code block 1  
}  
else if (condition2){  
    // code block 2  
}  
else {  
    // code block 3  
}
```

Ternary Statements

```
int number = -4;  
string result;  
// Using ternary operator  
result = (number > 0) ? "Positive Number!" : "Negative Number!";
```

Switch Case Statements

```
switch (expression) {  
    case x:  
        // code  
        break;  
    case y:  
        // code  
        break;  
    default:  
        // code  
}
```

C++ Loops/Iterative Statements

Loops are used when you want to do repetitive task in the program.

Types Of Loops

- While Loop
- For Loop
- Do-while Loop

While Loop

```
while(condition){  
    //code  
}
```

For Loop

```
for (init-statement; condition; final-expression) {  
    //code  
}
```

Multiple Variables in For Loop

```
for (int i=0, j=4; i<4, j>0; i++, j--) {  
    // code  
}
```

For Loop vs While Loop

```
// For Loop  
for (init-statement; condition; final-expression) {  
    //code  
}  
  
// while Loop  
init-statement  
while(condition) {  
    // code  
    final-expression  
}
```

Do-while Loop

```
do {  
    code  
} while(condition);
```

Function

How to declare a function.

```
returnType functionName(parameter1, parameter2){  
    // Statements  
}
```

Function Prototype

Like if you want define your function-A after the function-B but you want to call function-B in inside of function-A.

e.g.

```
#include<iostream>  
using namespace std;  
  
int add(int, int);    // prototype  
  
int main() {  
    cout<<add(2,3)<<endl;  
}  
  
int add(int a, int b){  
    return a+b;  
}
```