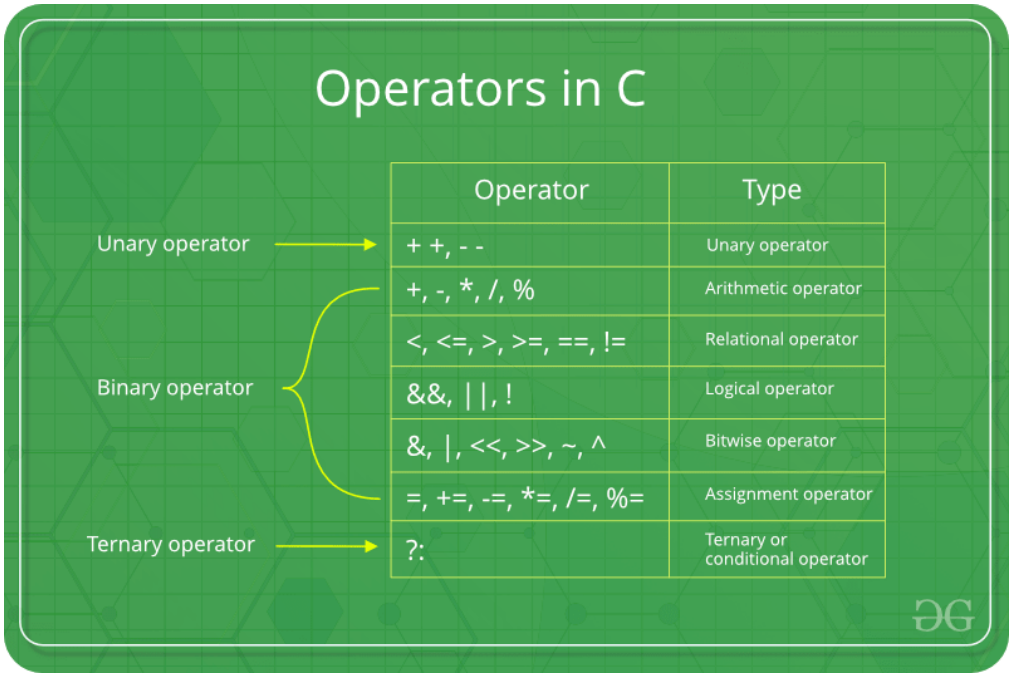
**Operators**

**Operators** are the foundation of any programming language. 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. For example, '+' is an operator used for addition, as shown below:

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

**C/C++ has many built-in operators and can be classified into 6 types:**

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operators
6. Other Operators




**1. Arithmetic Operators:**

These operators are used to perform arithmetic/mathematical operations on operands. Examples: (+, -, *, /, %,++,–). Arithmetic operators are of two types:

**a) Unary Operators**: Operators that operate or work with a single operand are unary operators. For example: Increment(++) and Decrement(–) Operators

```
int val = 5;
++val;  // 6
```

**b) Binary Operators**: Operators that operate or work with two operands are binary operators. For example: Addition(+), Subtraction(-), multiplication(*), Division(/) operators

```
int a = 7;
int b = 2;
cout<<a+b; // 9
```

**2. Relational Operators:**

These are used for the comparison of the values of two operands. For example, checking if one operand is equal to the other operand or not, whether an operand is greater than the other operand or not, etc. Some of the relational operators are (==, >= , <= )(See this article for more reference).

```
int a = 3;
int b = 5;
 a < b;
// operator to check if a is smaller than b
```

**3. Logical Operators:**

Logical Operators are used to combine two or more conditions/constraints or to complement the evaluation of the original condition in consideration. The result of the operation of a logical operator is a Boolean value either **true** or **false**.

For example, the **logical AND** represented as **'&&' operator in C or C++** returns true when both the conditions under consideration are satisfied. Otherwise, it returns false. Therefore, a && b returns true when both a and b are true (i.e. non-zero)(See this article for more reference).

```
(4 != 5) && (4 < 5);     // true
```

**4. Bitwise Operators:**

The Bitwise operators are used to perform bit-level operations on the operands. The operators are first converted to bit-level and then the calculation is performed on the operands. Mathematical operations such as addition, subtraction, multiplication, etc. can be performed at the bit-level for faster processing. For example, the **bitwise AND** represented as **& operator in C or C++** takes two numbers as operands and does AND on every bit of two numbers. The result of AND is 1 only if both bits are 1. (See this article for more reference).

```
int a = 5, b = 9;   // a = 5(00000101), b = 9(00001001)
cout << (a ^ b);   //  00001100
cout <<(~a);       // 11111010
```

**5. Assignment Operators:**

Assignment operators are used to assign value to a variable. The left side operand of the assignment operator is a variable and the right side operand of the assignment operator is a value. The value on the

right side must be of the same data type as the variable on the left side otherwise the compiler will raise an error.

**Different types of assignment operators are shown below:**

**a. "=":** This is the simplest assignment operator. This operator is used to assign the value on the right to the variable on the left.

For example:

```
a = 10;
b = 20;
ch = 'y';
```

```
// Operators in C++
#include<iostream>
using namespace std;

int main(){
   int a=10, b=5;
   // Arithmetic operators
   cout<<"Following are the Arithmetic operators in C++"<<endl;
   cout<<"The value of a + b is "<<a+b<<endl;
   cout<<"The value of a - b is "<<a-b<<endl;
   cout<<"The value of a * b is "<<a*b<<endl;
   cout<<"The value of a / b is "<<a/b<<endl;
   cout<<"The value of a % b is "<<a%b<<endl;
   cout<<"The value of a++ is "<<a++<<endl; // First print (a) and then increment it by 1
   cout<<"The value of a-- is "<<a--<<endl; // First print (a+1) and then decrease it by 1
   cout<<"The value of ++a is "<<++a<<endl; // Increment (a) by (a+1) and then print
   cout<<"The value of --a is "<<--a<<endl; // Decrement (a+1) by (a) and then print
   cout<<endl;

   // Assignment Operators --> used to assign values to variables
   // int a =3, b=9;
   // char d='d';

   // Comparison operators
   // Output of all these comparison operators will be (1) if it is true and (0) if it is false
   cout<<"Following are the comparison operators in C++"<<endl;
   cout<<"The value of a == b is "<<(a==b)<<endl;
   cout<<"The value of a != b is "<<(a!=b)<<endl;
   cout<<"The value of a >= b is "<<(a>=b)<<endl;
   cout<<"The value of a <= b is "<<(a<=b)<<endl;
   cout<<"The value of a > b is "<<(a>b)<<endl;
   cout<<"The value of a < b is "<<(a<b)<<endl;
   cout<<endl;
   // Logical operators
   cout<<"Following are the logical operators in C++"<<endl;
   cout<<"The value of this logical and operator ((a==b) && (a<b)) is:"<<((a==b) && (a<b))<<endl;
   cout<<"The value of this logical or operator ((a==b) || (a<b)) is:"<<((a==b) || (a<b))<<endl;
```

```
    cout<<"The value of this logical not operator (!(a==b)) is:"<<(!(a==b))<<endl;


    return 0;
}
```

**Output**

```
Following are the Arithmetic operators in C++
The value of a + b is 15
The value of a - b is 5
The value of a * b is 50
The value of a / b is 2
The value of a % b is 0
The value of a++ is 10
The value of a-- is 11
The value of ++a is 11
The value of --a is 10

Following are the comparison operators in C++
The value of a == b is 0
The value of a != b is 1
The value of a >= b is 1
The value of a <= b is 0
The value of a > b is 1
The value of a < b is 0

Following are the logical operators in C++
The value of this logical and operator ((a==b) && (a<b)) is:0
The value of this logical or operator ((a==b) || (a<b)) is:0
The value of this logical not operator (!(a==b)) is:1
```