

Introduction to C Plus Plus

Module 5 - Operators and Manipulators in C Plus Plus

Operators in C++

Assignment (=):

The assignment operator assigns a value to a variable $a=5$;

• This statement assigns the integer value 5 to the variable a. The part at the left of the assignment operator (=) is known as the lvalue (left value) and the right one as the rvalue (right value).

• The lvalue has to be a variable whereas the rvalue can be a constant, a variable, the result of an operation or any combination of these. The most important rule when assigning is the right-to-left rule:

• The assignment operation always takes place from right to left, and never the other way: $a= b$;

• This statement assigns to variable a (the lvalue) the value contained in variable b (the rvalue). The value that was stored until this moment in a is not considered at all in this operation, and in fact that value is lost.

Arithmetic operators (+, -, *, /, %)

The five arithmetical operations supported by the C++ language are: + addition - subtraction * multiplication / division % modulo

• Operations of addition, subtraction, multiplication and division literally correspond with their respective mathematical operators.

• The only one that you might not be so used to see may be modulo; whose operator is the percentage sign (%). Modulo is the operation that gives the remainder of a division of two values. For example, if we write: $a = 11 \% 3$; The variable a will contain the value 2, since 2 is the remainder from dividing 11 between 3.

Compound assignment (+=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=)

When we want to modify the value of a variable by performing an operation on the value currently stored in that variable we can use compound assignment operators:

• expression is equivalent to

• value += increase; value = value + increase;

• $a -= 5$; $a = a - 5$;

• $a /= b$; $a = a / b$;

• price *= units + 1; price = price * (units + 1);



CBSE



ICSE



NTSE



Banking &
Insurance



Central Govt.
Service



State Govt.
Services



LAW
Entrance



MBA
Entrance



Railways & Metro
Services

...many more

abhyasonline.in



**Course
&
Test Series**

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking &
Insurance**

 **Central Govt.
Service**

 **State Govt.
Services**

 **LAW
Entrance**

 **MBA
Entrance**

 **Railways & Metro
Services**

...many more

abhyasonline.in

Introduction to 'C++' Language

Increase and decrease (++, --)

- Shortening even more some expressions, the increase operator (++) and the decrease operator (--) increase or reduce by one the value stored in a variable.
- They are equivalent to +=1 and to -=1, respectively.
- Thus: c++; c+=1; c=c+1; are all equivalent in its functionality: the three of them increase by one the value of c.
- A characteristic of this operator is that it can be used both as a prefix and as a suffix. That means that it can be written either before the variable identifier (c++) or after it (c++).

Relational and equality operators (==, !=, >, <, >=, <=):

In order to evaluate a comparison between two expressions we can use the relational and equality operators.

- The result of a relational operation is a Boolean value that can only be true or false, according to its Boolean result.
- We may want to compare two expressions, for example, to know if they are equal or if one is greater than the other is.

Here is a list of the relational and equality operators that can be used in C++:

- == Equal to
- != Not equal to
- > Greater than
- < Less than
- >= Greater than or equal to
- <= Less than or equal to

Here there are some examples:

- (7 == 5) // evaluates to false.
- (5 > 4) // evaluates to true.
- (3 != 2) // evaluates to true.
- (6 >= 6) // evaluates to true.
- (5 < 5) // evaluates to false.

Of course, instead of using only numeric constants, we can use any valid expression, including variables.

Suppose that

- a=2, b=3 and c=6, (a == 5) // evaluates to false since a is not equal to 5.
- (a*b >= c) // evaluates to true since (2*3 >= 6) is true.
- (b+4 > a*c) // evaluates to false since (3+4 > 2*6) is false.
- ((b=2) == a) // evaluates to true.



CBSE



ICSE



NTSE



Banking &
Insurance



Central Govt.
Service



State Govt.
Services



LAW
Entrance



MBA
Entrance



Railways & Metro
Services

...many more

abhyasonline.in

Conditional operator (?)

The conditional operator evaluates an expression returning a value if that expression is true and a different one if the expression is evaluated as false.

- Its format is: condition ? result1 : result2
- If condition is true the expression will return result1, if it is not it will return result2.
- $7==5 ? 4 : 3$ // returns 3, since 7 is not equal to 5.
- $7==5+2 ? 4 : 3$ // returns 4, since 7 is equal to 5+2.
- $5>3 ? a : b$ // returns the value of a, since 5 is greater than 3.
- $a>b ? a : b$ // returns whichever is greater, a or b.

Comma operator (,):

The comma operator (,) is used to separate two or more expressions that are included where only one expression is expected.

- When the set of expressions has to be evaluated for a value, only the rightmost expression is considered.
- For example, the following code: $a = (b=3, b+2)$; Would first assign the value 3 to b, and then assign b+2 to variable a.
- So, at the end, variable a would contain the value 5 while variable b would contain value 3.

sizeof()

This operator accepts one parameter, which can be either data type or a variable itself and returns the size in bytes of that type or object:

- $a = \text{sizeof}(\text{char})$; This will assign the value 1 to a, because char is a one-byte long type.
- The value returned by sizeof is a constant, so it is always determined before program execution.

Scope Resolution Operator (::)

The **scope resolution operator (::)** in C++ is used to define the scope or context in which a function or variable exists. It helps you access variables, functions, or classes that are defined outside the current scope or class.

- :: is used to clarify which function, variable, or class you're referring to when there are multiple definitions or scopes involved.
- It's especially useful for accessing global variables, static class members, and functions outside of classes.



Manipulators in C++

In C++, **manipulators** are special functions used to format the output or input in a specific way. They help in controlling how data is displayed on the screen (output) or how input is taken.

Common Manipulators in C++:

1. **endl (End of Line):**

Purpose: It inserts a new line and flushes the output buffer (clears the stream).

Usage: It moves the cursor to the next line in the output.

```
std::cout << "Hello, world!" << std::endl; // Moves to the next line after printing
```

2. **setw() (Set Width):**

Purpose: It sets the width for the next output. If the output is smaller than the specified width, spaces are added to the left to make it the correct width.

Usage: It helps align the output neatly in columns.

```
#include <iomanip> // Required for setw()
```

```
std::cout << std::setw(10) << 123 << std::endl; // Will print '    123' (10 spaces total)
```

Assignment

Ques 1: What is the output of the following code?

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    int a = 10, b = 5;  
    cout << a + b << endl;  
    cout << a - b << endl;  
    cout << a * b << endl;  
    cout << a / b << endl;  
    return 0;  
}
```



CBSE



ICSE



NTSE



Banking &
Insurance



Central Govt.
Service



State Govt.
Services



LAW
Entrance



MBA
Entrance



Railways & Metro
Services

...many more

abhyasonline.in



Course
&
Test Series

Introduction to 'C++' Language

Ques 2: What is the output of the following code?

```
#include <iostream>
using namespace std;

int main() {
    int x = 20, y = 30;
    cout << (x == y) << endl;
    cout << (x != y) << endl;
    return 0;
}
```

Ques 3: What will be the output of the following code?

```
#include <iostream>
using namespace std;

int main() {
    bool a = true, b = false;
    cout << (a && b) << endl;
    cout << (a || b) << endl;
    return 0;
}
```



CBSE



ICSE



NTSE



Banking &
Insurance



Central Govt.
Service



State Govt.
Services



LAW
Entrance



MBA
Entrance



Railways & Metro
Services



...many more

abhyasonline.in

