C++ Header files & Operators | C++ Tutorials for Beginners #6

▶

Overview　　**Q&A**　　**Files**　　**Announcements**

C++ Header files & Operators | C++ Tutorials for Beginners #6

In this C++ tutorial, we will talk about header files and operators. In our last lesson, we discussed the basic input and output. Lets now cover header files and operators in C++ language:

Header Files in C++

"**#include**" is used in C++ to import header files in our C++ program. The reason to introduce the **"<iostream>"** header file into our program is that functions like **"cout"** and **"cin"** are defined in that header file. There are two types of header files:

System Header Files

System header files ships with the compiler. For example, "**#include <iostream>**". To see the references for header files click here

User-Defined Header Files

The programmer writes User-defined header files himself. To include your header file in the program, you first need to make a header file in the current directory, and then you can add it.

Operators in C++

Operators are used for producing output by performing various types of calculations on two or more inputs. In this lecture, we will see the operators in C++.

Arithmetic Operators

Arithmetic operators are used for performing mathematical operations like (+, -, *). The arithmetic operators are shown in Figure 1.

Course Content

Hide Player

1. Introduction to C++, Installing VS Code, g++ & more | C++ Tutorials for Beginners #1
▶ Free YouTube Video

2. Basic Structure of a C++ Program | C++ Tutorials for Beginners #2
▶ Free YouTube Video

3. Variables & Comments in C++ in Hindi | C++ Tutorials for Beginners #3
▶ Free YouTube Video

4. Variable Scope & Data Types in C++ in Hindi | C++ Tutorials for Beginners #4
▶ Free YouTube Video

5. C++ Basic Input/Output & More | C++ Tutorials for Beginners #5
▶ Free YouTube Video

6. C++ Header files & Operators | C++ Tutorials for Beginners #6
▶ Free YouTube Video

```
int a=4, b=5;
cout<<"Operators in C++:"<<endl;
cout<<"Following are the types of operators in C++"<<endl;
// Arithmetic operators
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;
cout<<"The value of a-- is "<<a--<<endl;
cout<<"The value of ++a is "<<++a<<endl;
cout<<"The value of --a is "<<--a<<endl;
cout<<endl;
```

Figure 1: Arithmetic Operators

1. The function **"a+b"**, will add a and b values and print them.
2. The function **"a-b "** will subtract a and b values and print them.
3. The function **"a*b"** will multiply a and b values and print them.
4. The function **"a/b "**, will divide a and b values and print them.
5. The function **"a%b "**, will take the modulus of a and b values and print them.
6. The function **"a++"** will first print the value of a and then increment it by 1.
7. The function **"a--"**, will first print the value of a and then decrement it by 1.
8. The function **"++a"**, will first increment it by one and then print its value.
9. The function **"--a"**, will first decrement it by one and then print its value.

The output of these arithmetic operators is shown in figure 2.

```
Following are the types of operators in C++
The value of a + b is 9
The value of a - b is -1
The value of a * b is 20
The value of a / b is 0
The value of a % b is 4
The value of a ++ is 4
The value of a -- is 5
The value of ++a is 5
The value of --a is 4
```

Figure 2: Arithmetic Operators Output

- ## Assignment Operators

Assignment operators are used for assigning values to variables. For example: **int a = 10, b = 5;**

- ## Comparison Operators

Comparison operators are used for comparing two values. Examples of comparison operators are shown in figure 3.

```
// Comparison operators
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;
```

Figure 3: Comparison Operators

1. The function **"(a==b)"**, will compare a and b values and check if they are equal. The output will be one if equal, and 0 if not.
2. The function **"(a!=b)"**, will compare a and b values and check if "a" is not equal to "b". The output will be one if not equal and 0 if equal.
3. The function **"(a>=b)"**, will compare a and b values and check if "a" is greater than or equal to "b". The output will be one if greater or equal, and 0 if not.

4. The function **"(a<=b)"**, will compare a and b values and check if "b" is greater than or equal to "a". The output will be one if greater or equal, and 0 if not.

5. The function **"(a>b)"**, will compare a and b values and check if "a" is greater than "b". The output will be one if greater and 0 if not.

6. The function **"(a<b)"**, will compare a and b values and check if "b" is greater than "a". The output will be one if greater and 0 if not.

The output of these comparison operators is shown in figure 4.

```
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 0
The value of a <= b is 1
The value of a > b is 0
The value of a < b is 1
```

Figure 4: Comparison Operators Output

- Logical Operators

Logical operators are used for comparing two expressions. For example ((a==b) && (a>b)). More examples of logical operators are shown in figure 5.

```
// 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;
```

Figure 5: Logical Operators

1. The function **"((a==b)&& (a<b))"** will first compare a and b values and check if they are equal or not; if they are equal, the next expression will check whether "a" is smaller than "b". The output will be one if both expressions are true and 0 if not.

2. The function **"((a==b) || (a<b))"**, will first compare a and b values and check if they are equal or not, even if they are not equal it will still check the next expression ie whether "a" is smaller than "b" or not. The output will be one if any one of the expressions is true and 0 if both are false.

3. The function **"(!(a==b))"**, will first compare a and b values and check if they are equal or not. The output will be inversed ie if "a" and "b" are equal; the output will be 0 and 1 otherwise.

The output of these logical operators is shown in figure 6.

```
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:1
The value of this logical not operator (!(a==b)) is:1
```

Figure 6: Logical Operators Output

That's it for this tutorial. In this lecture, we have covered some important operators in C++ language, but there are still some operators left, which we will cover in upcoming tutorials.

Code as described/written in the video

```cpp
    // Assignment Operators --> used to assign values to variables

    // int a =3, b=9;

    // char d='d';


    // Comparison operators

    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;


    // 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;

}
```

Copied!

← Previous   Next →