C Operators

Topics : C
Written on April 12, 2023

In C programming language, operators are used to perform operations on operands. An operand is a variable or a value on which the operator operates. C programming language supports various types of operators, such as arithmetic operators, relational operators, logical operators, bitwise operators, assignment operators, and conditional operators.

Here are the most commonly used operators in C programming language:

1. **Arithmetic Operators**: Arithmetic operators are used to perform arithmetic operations on numeric values.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (remainder after division)

Example:

```
int a = 10, b = 5, c;
c = a + b; // c is now 15
c = a - b; // c is now 5
c = a * b; // c is now 50
c = a / b; // c is now 2
c = a % b; // c is now 0
```

2. **Relational Operators**: Relational operators are used to compare two values.

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

Example:

```
int a = 10, b = 5;
if (a == b) {
    printf("a is equal to b\n");
}
if (a > b) {
    printf("a is greater than b\n");
}
if (a < b) {
    printf("a is less than b\n");
}
```

3. **Logical Operators**: Logical operators are used to perform logical operations on boolean values.

Operator	Description
&&	Logical AND
||	Logical OR
!	Logical NOT

Example:

```
int a = 10, b = 5, c = 15;
if (a > b && c > a) {
    printf("Both conditions are true\n");
}
if (a > b || c < a) {
    printf("At least one condition is true\n");
}
if (!(a > b)) {
    printf("a is not greater than b\n");
}
```

4. **Bitwise Operators**: Bitwise operators are used to perform bitwise operations on binary values.

Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise NOT
<<	Left shift
>>	Right shift

Example:

```
unsigned int a = 60;  // 0011 1100
unsigned int b = 13;  // 0000 1101
```

```
unsigned int c;
c = a & b;   // 0000 1100
c = a \| b;   // 0011 1101
c = a ^ b;   // 0011 0001
c = ~a;      // 1100 0011
c = a << 2;  // 1111 0000
c = a >> 2;  // 0000 1111
```

5. **Assignment Operators**: Assignment operators are used to assign values to variables.

Operator	Description
=	Assignment
+=	Add and assign

```
int x = 10;
```