

www.programming9.com

C Programming Tutorials

OPERATORS

N.V.RAJA SEKHAR REDDY

What is an operator?

- ▶ An operator is a symbol that tells the computer to perform certain mathematical or logical manipulations.
- ▶ These operators are used in programs to manipulate data and variables.

Types of Operators

1. Arithmetic operators
2. Relational operators
3. Logical operators
4. Assignment operators
5. Increment and decrement operators
6. Conditional operators
7. Bitwise operators
8. Special operators

ARITHMETIC OPERATORS:

- ▶ Arithmetic operators are used to perform numerical calculations among the values.

OPERATOR	MEANING
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulo Division

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int a, b, add, sub, mul, div, rem;
```

```
    printf("Enter a, b values : ");
```

```
    scanf("%d%d",&a,&b);
```

```
                                // Reading two values
```

```
    add=a+b;
```

```
                                // Addition Operator
```

```
    sub=a-b;
```

```
                                // Subtraction Operator
```

```
    mul=a*b;
```

```
                                // Multiplication Operator
```

```
    div=a/b;
```

```
                                // Division Operator
```

```
    rem=a%b;
```

```
                                // Remainder (Modulo) Operator
```

```
    printf("Result of addition is=%d\n", add);
```

```
    printf("Result of subtraction=%d\n", sub);
```

```
    printf("Result of multiplication is=%d\n", mul);
```

```
    printf("Result of division is=%d\n", div);
```

```
    printf("Result of remainder=%d\n",rem);
```

```
    return 0;    }
```

RELATIONAL OPERATOR:

- ▶ Relational Operators are used to compare two quantities and take certain decision depending on their relation.

If the specified relation is true it returns one.

If the specified relation is false it returns zero.

OPERATOR	MEANING
<	Is less than
<=	Is less than or equal to
>	Is greater than
>=	Is greater than or equal to
==	Is equal to
!=	Is not equal to

Example Program

```
#include<stdio.h>

void main()
{
    int a, b;
    printf("Enter values for a and b : ");
    scanf("%d %d", &a, &b);
    printf("\n The < value of a is %d", a<b);
    printf("\n The <= value of a is %d", a<=b);
    printf("\n The > value of a is %d", a>b);
    printf("\n The >= value of a is %d", a>=b);
    printf("\n The == value of a is %d", a==b);
    printf("\n The != value of a is %d", a!=b);
}
```

LOGICAL OPERATORS:

- ▶ These operators are used for testing more than one condition and making decisions.

'c' has three logical operators they are:

OPERATOR	MEANING
&&	Logical AND
 	Logical OR
!	Logical NOT

Logical Operators Example Program

```
#include<stdio.h>

void main()
{
    int a, b;
    printf("Enter values for a and b : ");
    scanf("%d %d", &a, &b);
    printf("\n %d", (a<b)&&(a!=b));
    printf("\n %d", (a<b)|| (b<a));
    printf("\n %d", !(a==b));
}
```

ASSIGNMENT OPERATORS

- ▶ These operators are used for assigning the result of an expression to a variable.
- ▶ `b=a;`

OPERATORS:

`==, +=, -=, *=, /=, %=,`
`>>=, <<=, &=, |=, ^=`

Assignment Operators Example

```
#include<stdio.h>

void main()
{
    int a, b, c;
    printf("Enter the values for a and b : ");
    scanf("%d %d",&a,&b);
    printf("\n the values of= is:%d",c=a+b);
    printf("\n the values of +=is:%d",c+=b);
    printf("\n the value of-= is:%d",c-=a);
    printf("\n the value of *=is:%d",c*=a);
    printf("\n the value of /=is:%d",c/=b);
    printf("\n the value of %=is:%d",c%=b);
}
```

INCREMENT & DECREMENT OPERATORS

Two most useful operators which are present in 'c' are increment and decrement operators.

Operators: ++ and --

The operator ++ adds one to the operand

The operator -- subtracts one from the operand.

Both are unary operators and can be used as pre or post increment/decrement.

INCREMENT & DECREMENT OPERATORS

EXAMPLE

```
#include<stdio.h>

void main()
{
    int a,b,c;
    printf("Enter the values for a and b :");
    scanf("%d %d", &a, &b);
    printf("\n The value of c is %d", c=++a);
    printf("\n The value of c is %d", c=a++);
    printf("\n The value of c is %d", c=--b);
    printf("\n The value of c is %d", c=b--);
}
```

CONDITIONAL OPERATORS

These conditional operator are used to construct conditional expressions of the form.

Syntax:

`exp1?exp2:exp3.`

where `exp1,exp2,exp3` are expressions.

Operator: `?:` (ternary operator)

Conditional Operators Example

```
#include<stdio.h>

void main()
{
    int a, b, x;
    printf("Enter the values of a add b : ");
    scanf("%d %d", &a, &b);
    x=(a>b)?a:b;
    printf("Biggest Value is :%d",x);
}
```

BITWISE OPERATORS

- ▶ These operators work on bit level
- ▶ Applied to Integers only

OPERATOR	MEANING
&	Bitwise AND
	Bitwise OR
^	Bitwise Exclusive OR
<<	Shift Left
>>	Shift Right

SPECIAL OPERATORS

'C' supports some special operators such as comma operator, sizeof operator and pointer operators.

Comma operator:

the comma operator is used to combine related expressions.

A comma linked list of expressions are evaluated left to right and the value of right most expression is the value of combined expression..

Example: `value=(x=10, y=5, x+y);`

Special Operators Contd...

Sizeof Operator:

Sizeof is an operator used to return the number of bytes the operand occupies.

Syntax:

```
m=sizeof(sum);  
k=sizeof(2351);
```

www.programming9.com

Like us @

www.facebook.com/programming9