

# C# Operators Precedence with Examples

In **C#**, **Operators Precedence** is useful to define multiple operators in a single expression, and the evaluation of expression can be happened based on the priority of operators.

For example, the multiplication operator (\*) is having higher precedence than the addition operator (+). So if we use both multiplication (\*) and addition (+) operators in a single expression, first, it will evaluate the multiplication part and then the addition part in expression.

Following is the simple example of defining an expression with operator precedence in **C#**.

```
int i = 3 + 4 * 5;
```

When we execute the above statement, the multiplication part (**4 \* 5**) will be evaluated first. After that, the addition part (**3 + 12**) will be executed, and the **i** value will become a **23**.

As said earlier, the multiplication (\*) operator is having higher precedence than the addition (+) operator so that the first multiplication part will be executed.

The following table lists the different types of operators available in **C#** relational operators.

Category	Operator(s)
Postfix / Prefix	++, --
Unary	+, -, !, ~
Multiplicative	*, /, %
Additive	+, -
Shift	<<, >>
Relational	<, <=, >, >=
Equality	==, !=
Bitwise	&,  , ^
Logical	&&,
Conditional	?:
Assignment	=, +=, -=, *=, /=, %=, &=,  =, ^=, <<=, >>=

## C# Operator Precedence Example

Following is the example of implementing operator precedence in the **C#** programming language.

```
using System;

namespace Tutlane
{
    class Program
    {
        static void Main(string[] args)
        {
            int x = 20, y = 5, z = 4;
            int result = x / y + z;
            Console.WriteLine("Result1: "+result);
            bool result2 = z <= y + x;
            Console.WriteLine("Result2: "+result2);
            Console.WriteLine("Press Enter Key to Exit..");
        }
    }
}
```

```
Console.ReadLine());
```

```
}
```

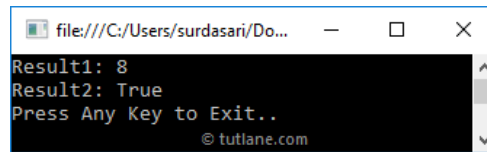
```
}
```

```
}
```

If you observe the above example, we implemented operator precedence with multiple operators. The evaluation of expressions will be done based on operators' priority in the c# programming language.

## Output of C# Operators Precedence Example

When we execute the above c# program, we will get the result as shown below.



```
file:///C:/Users/surdasari/Do...
Result1: 8
Result2: True
Press Any Key to Exit..
© tutlane.com
```

This is how we can implement operator precedence in the c# programming language based on our requirements.

### CONTACT US

📍 **Address:** No.1-93, Pochamma Colony, Manikonda, Hyderabad, Telangana - 500089

✉ **Email:** support@tutlane.com (mailto:support@tutlane.com)