

# C# - BITWISE OPERATORS

[http://www.tutorialspoint.com/csharp/csharp\\_bitwise\\_operators.htm](http://www.tutorialspoint.com/csharp/csharp_bitwise_operators.htm)

Copyright © tutorialspoint.com

The Bitwise operators supported by C# are listed in the following table. Assume variable A holds 60 and variable B holds 13, then:

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	<b>A &amp; B</b> = 12, which is 0000 1100
	Binary OR Operator copies a bit if it exists in either operand.	A   B = 61, which is 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	A ^ B = 49, which is 0011 0001
~	Binary Ones Complement Operator is unary and has the effect of 'flipping' bits.	A = 61, which is 1100 0011 in 2's complement due to a signed binary number.
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 = 240, which is 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 = 15, which is 0000 1111

## Example

The following example demonstrates all the bitwise operators available in C#:

```
using System;
namespace OperatorsApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            int a = 60;           /* 60 = 0011 1100 */
            int b = 13;           /* 13 = 0000 1101 */
            int c = 0;

            c = a & b;             /* 12 = 0000 1100 */
            Console.WriteLine("Line 1 - Value of c is {0}", c );

            c = a | b;             /* 61 = 0011 1101 */
            Console.WriteLine("Line 2 - Value of c is {0}", c);

            c = a ^ b;             /* 49 = 0011 0001 */
            Console.WriteLine("Line 3 - Value of c is {0}", c);

            c = ~a;                /* -61 = 1100 0011 */
            Console.WriteLine("Line 4 - Value of c is {0}", c);

            c = a << 2;            /* 240 = 1111 0000 */
            Console.WriteLine("Line 5 - Value of c is {0}", c);

            c = a >> 2;            /* 15 = 0000 1111 */
            Console.WriteLine("Line 6 - Value of c is {0}", c);
            Console.ReadLine();
        }
    }
}
```

```
}  
}  
}
```

When the above code is compiled and executed, it produces the following result:

```
Line 1 - Value of c is 12  
Line 2 - Value of c is 61  
Line 3 - Value of c is 49  
Line 4 - Value of c is -61  
Line 5 - Value of c is 240  
Line 6 - Value of c is 15
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js