

# JUMP STATEMENTS IN C#

The break, goto, continue, return and throw statements are known as jump statements. These are used to transfer program control from one point in the program to another point, at any time. Let's understand how these work?

## break statement

This statement terminates the execution of loop or switch in which it appears and transfers program control to the next statement which is placed immediately after the loop or switch.

```
1. public class Example
2. {
3.     static void Main(string[] args)
4.     {
5.         for (int i = 1; i <= 10; i++)
6.         {
7.             if (i == 5)
8.             {
9.                 break;
10.            }
11.            Console.WriteLine(i);
12.        }
13.        Console.WriteLine("Next statement placed after loop");
14.    }
15. }
16. /* Output:
17.  1
18.  2
19.  3
20.  4
21.  Next statement placed after loop
22. */
```

This statement is also used to terminate an inner nested loop, and return control to the outer loop.

## goto statement

This statement transfers program control to a labeled statement. The label statement must exist in the scope of the goto statement. More than one goto statement can transfer control to the same label. This statement can be used to get out from a loop or an inner nested loop to outer loop.

```
1. public class Example
2. {
3.     static void Main(string[] args)
4.     {
5.         for (int i = 1; i <= 10; i++)
6.         {
7.             if (i == 5)
8.             {
9.                 goto Exitlabel;
10.            }
11.            Console.WriteLine(i);
12.        }
13.        Console.WriteLine("Next statement placed after loop");
14.
15.        Exitlabel: //goto label
16.        Console.WriteLine("Labeled statement");
17.    }
18. }
19. /* Output:
20.  1
21.  2
22.  3
23.  4
24.  Labeled statement
25. */
```

Unlike break statement, it does not transfer the program control to next statement which is placed immediately after the loop or switch.

You can also use goto statement to transfer control to a specific switch-case label or the default label in a switch statement.

It is not recommended to use goto statement since this makes the program logic complex and difficult to understand. It also becomes difficult to trace the control flow of a program execution.

## continue statement

This statement skips the current iteration and passes program control to the next iteration of the enclosing loop in which it appears.

```
1. public class Example
2. {
3.     static void Main(string[] args)
4.     {
5.         for (int i = 1; i <= 10; i++)
6.         {
7.             if (i <= 5)
8.             {
9.                 continue;
10.            }
11.            Console.WriteLine(i);
12.        }
13.        Console.WriteLine("Next statement placed after loop");
14.    }
15. }
16. /* Output:
17.  6
18.  7
19.  8
20.  9
21.  10
22.  Next statement placed after loop
23. */
```

Unlike break statement, it does not terminate the loop execution but it skip the current iteration of the loop and passes program control to the next iteration of the enclosing loop.

## return statement

This statement terminates the execution of the method in which it appears and returns control to the calling method.

```
1. public class Example
2. {
3.     static void Main(string[] args)
4.     {
5.         double length = 5.0, width = 2.5;
6.
7.         double result = CalculateArea(length, width);
8.         Console.WriteLine("The area is {0:0.00}", result);
9.     }
10.    public static double CalculateArea(double length, double width)
11.    {
12.        double area = length * width;
13.        return area;
14.    }
15. }
16. /* Output:
17.    The area is 12.50
18. */
```

If the return statement appear in a try block and the finally block is also exists, it finally block will be executed before control returns to the calling method.

## throw statement

This statement throws an exception which indicate that an error has occurred during the program execution. This statement is used with a combination of try-catch or try-finally statements.

```
1. public class Example
2. {
3.     static void Main(string[] args)
4.     {
5.         try
6.         {
7.             int number = 5, x = 0;
```

```
8.
9.  double result = number / x;
10. Console.WriteLine("Result is {0}", result);
11. }
12. catch (Exception ex)
13. {
14.  Console.WriteLine("Exception: "+ex.Message);
15.  throw ex;
16. }
17. }
18. }
19. /* Output:
20. Exception: Attempted to divide by zero.
21. */
```

Source : <http://www.dotnet-tricks.com/Tutorial/csharp/068a171113-Jump-statements-in-C#.html>