**C# Control Statement**

# C# if-else

In C# programming, the *if statement* is used to test the condition. There are various types of if statements in C#.

* if statement
* if-else statement
* nested if statement
* if-else-if ladder

## C# IF Statement

The C# if statement tests the condition. It is executed if condition is true.

**Syntax**

1. **if**(condition){
2. //code to be executed
3. }
4. **if**(condition){
5. //code to be executed
6. }

### C# If Example

1. using System;
2. **public** **class** IfExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. **int** num = 10;
7. **if** (num % 2 == 0)
8. {
9. Console.WriteLine("It is even number");
10. }
12. }
13. }

Output:

It is even number

## C# IF-else Statement

**Syntax:**

1. **if**(condition){
2. //code if condition is true
3. }**else**{
4. //code if condition is false
5. }

The C# if-else statement also tests the condition. It executes the if block if condition is true otherwise else block is executed.

### C# If-else Example

1. using System;
2. **public** **class** IfExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. **int** num = 11;
7. **if** (num % 2 == 0)
8. {
9. Console.WriteLine("It is even number");
10. }
11. **else**
12. {
13. Console.WriteLine("It is odd number");
14. }
16. }
17. }

Output:

It is odd number

## C# If-else Example: with input from user

In this example, we are getting input from the user using **Console.ReadLine()** method. It returns string. For numeric value, you need to convert it into int using **Convert.ToInt32()** method.

1. using System;
2. **public** **class** IfExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. Console.WriteLine("Enter a number:");
7. **int** num = Convert.ToInt32(Console.ReadLine());
9. **if** (num % 2 == 0)
10. {
11. Console.WriteLine("It is even number");
12. }
13. **else**
14. {
15. Console.WriteLine("It is odd number");
16. }
18. }
19. }

Output:

Enter a number:11

It is odd number

Output:

Enter a number:12

It is even number

## C# IF-else-if ladder Statement

The C# if-else-if ladder statement executes one condition from multiple statements.

**Syntax:**

1. **if**(condition1){
2. //code to be executed if condition1 is true
3. }**else** **if**(condition2){
4. //code to be executed if condition2 is true
5. }
6. **else** **if**(condition3){
7. //code to be executed if condition3 is true
8. }
9. ...
10. **else**{
11. //code to be executed if all the conditions are false
12. }

### C# If else-if Example

1. using System;
2. **public** **class** IfExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. Console.WriteLine("Enter a number to check grade:");
7. **int** num = Convert.ToInt32(Console.ReadLine());
9. **if** (num <0 || num >100)
10. {
11. Console.WriteLine("wrong number");
12. }
13. **else** **if**(num >= 0 && num < 50){
14. Console.WriteLine("Fail");
15. }
16. **else** **if** (num >= 50 && num < 60)
17. {
18. Console.WriteLine("D Grade");
19. }
20. **else** **if** (num >= 60 && num < 70)
21. {
22. Console.WriteLine("C Grade");
23. }
24. **else** **if** (num >= 70 && num < 80)
25. {
26. Console.WriteLine("B Grade");
27. }
28. **else** **if** (num >= 80 && num < 90)
29. {
30. Console.WriteLine("A Grade");
31. }
32. **else** **if** (num >= 90 && num <= 100)
33. {
34. Console.WriteLine("A+ Grade");
35. }
36. }
37. }

Output:

Enter a number to check grade:66

C Grade

Output:

Enter a number to check grade:-2

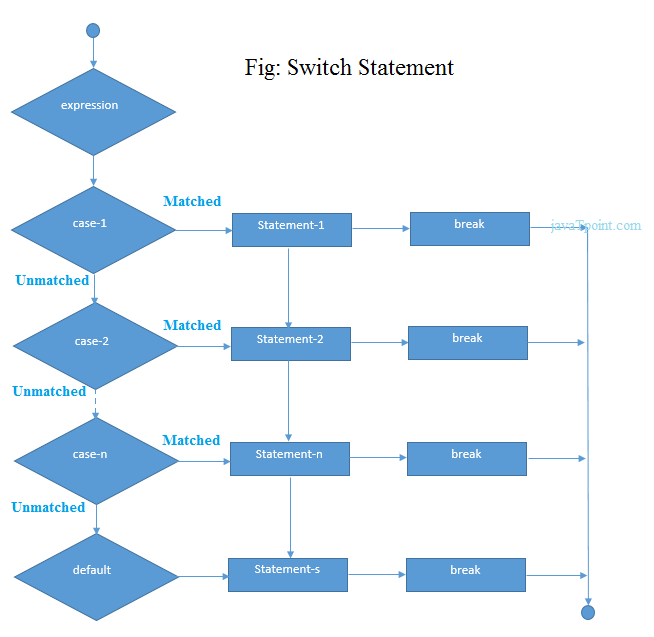
wrong number

# C# switch

The C# switch statement executes one statement from multiple conditions. It is like if-else-if ladder statement in C#.

**Syntax:**

1. **switch**(expression){
2. **case** value1:
3. //code to be executed;
4. **break**;
5. **case** value2:
6. //code to be executed;
7. **break**;
8. ......
10. **default**:
11. //code to be executed if all cases are not matched;
12. **break**;
13. }



### C# Switch Example

1. using System;
2. **public** **class** SwitchExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. Console.WriteLine("Enter a number:");
7. **int** num = Convert.ToInt32(Console.ReadLine());
9. **switch** (num)
10. {
11. **case** 10: Console.WriteLine("It is 10"); **break**;
12. **case** 20: Console.WriteLine("It is 20"); **break**;
13. **case** 30: Console.WriteLine("It is 30"); **break**;
14. **default**: Console.WriteLine("Not 10, 20 or 30"); **break**;
15. }
16. }
17. }

Output:

2.3M

250

History of Java

Enter a number:

10

It is 10

Output:

Enter a number:

55

Not 10, 20 or 30

#### Note: In C#, break statement is must in switch cases.

# C# For Loop

The C# for loop is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop than while or do-while loops.

The C# for loop is same as C/C++. We can initialize variable, check condition and increment/decrement value.

**Syntax:**

2.1M

253

Exception Handling in Java - Javatpoint

**Next**

**Stay**

1. **for**(initialization; condition; incr/decr){
2. //code to be executed
3. }

**Flowchart:**



### C# For Loop Example

1. using System;
2. **public** **class** ForExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. **for**(**int** i=1;i<=10;i++){
7. Console.WriteLine(i);
8. }
9. }
10. }

Output:

1

2

3

4

5

6

7

8

9

10

## C# Nested For Loop

In C#, we can use for loop inside another for loop, it is known as nested for loop. The inner loop is executed fully when outer loop is executed one time. So if outer loop and inner loop are executed 3 times, inner loop will be executed 3 times for each outer loop i.e. total 9 times.

Let's see a simple example of nested for loop in C#.

1. using System;
2. **public** **class** ForExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. **for**(**int** i=1;i<=3;i++){
7. **for**(**int** j=1;j<=3;j++){
8. Console.WriteLine(i+" "+j);
9. }
10. }
11. }
12. }

Output:

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

3 3

## C# Infinite For Loop

If we use double semicolon in for loop, it will be executed infinite times. Let's see a simple example of infinite for loop in C#.

1. using System;
2. **public** **class** ForExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. **for** (; ;)
7. {
8. Console.WriteLine("Infinitive For Loop");
9. }
10. }
11. }

Output:

Infinitive For Loop

Infinitive For Loop

Infinitive For Loop

Infinitive For Loop

Infinitive For Loop

ctrl+c

# C# While Loop

In C#, while loop is used to iterate a part of the program several times. If the number of iteration is not fixed, it is recommended to use while loop than for loop.

**Syntax:**

1. **while**(condition){
2. //code to be executed
3. }

**Flowchart:**

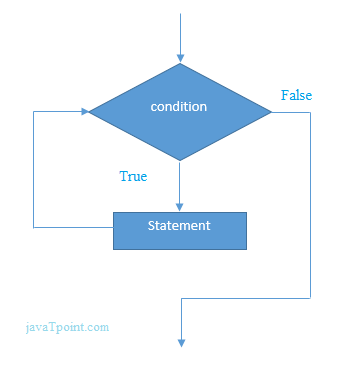
2.1M

253

Exception Handling in Java - Javatpoint

**Next**

**Stay**



### C# While Loop Example

Let's see a simple example of while loop to print table of 1.

1. using System;
2. **public** **class** WhileExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. **int** i=1;
7. **while**(i<=10)
8. {
9. Console.WriteLine(i);
10. i++;
11. }
12. }
13. }

Output:

1

2

3

4

5

6

7

8

9

10

### C# Nested While Loop Example:

In C#, we can use while loop inside another while loop, it is known as nested while loop. The nested while loop is executed fully when outer loop is executed once.

Let's see a simple example of nested while loop in C# programming language.

1. using System;
2. **public** **class** WhileExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. **int** i=1;
7. **while**(i<=3)
8. {
9. **int** j = 1;
10. **while** (j <= 3)
11. {
12. Console.WriteLine(i+" "+j);
13. j++;
14. }
15. i++;
16. }
17. }
18. }

Output:

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

3 3

### C# Infinitive While Loop Example:

We can also create infinite while loop by passing true as the test condition.

1. using System;
2. **public** **class** WhileExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. **while**(**true**)
7. {
8. Console.WriteLine("Infinitive While Loop");
9. }
10. }
11. }

Output:

Infinitive While Loop

Infinitive While Loop

Infinitive While Loop

Infinitive While Loop

Infinitive While Loop

ctrl+c

# C# Do-While Loop

The C# do-while loop is used to iterate a part of the program several times. If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use do-while loop.

The C# do-while loop is executed at least once because condition is checked after loop body.

**Syntax:**

2.1M

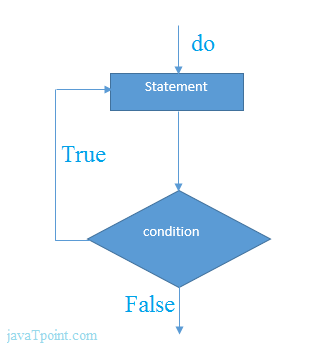
253

Exception Handling in Java - Javatpoint

**Next**

**Stay**

1. **do**{
2. //code to be executed
3. }**while**(condition);



### C# do-while Loop Example

Let's see a simple example of C# do-while loop to print the table of 1.

1. **using** System;
2. **public** **class** DoWhileExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
6. **int** i = 1;
8. **do**{
9. Console.WriteLine(i);
10. i++;
11. } **while** (i <= 10) ;
13. }
14. }

Output:

1

2

3

4

5

6

7

8

9

10

## C# Nested do-while Loop

In C#, if you use do-while loop inside another do-while loop, it is known as nested do-while loop. The nested do-while loop is executed fully for each outer do-while loop.

Let's see a simple example of nested do-while loop in C#.

1. **using** System;
2. **public** **class** DoWhileExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
6. **int** i=1;
8. **do**{
9. **int** j = 1;
11. **do**{
12. Console.WriteLine(i+" "+j);
13. j++;
14. } **while** (j <= 3) ;
15. i++;
16. } **while** (i <= 3) ;
17. }
18. }

Output:

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

3 3

## C# Infinitive do-while Loop

In C#, if you pass **true** in the do-while loop, it will be infinitive do-while loop.

**Syntax:**

1. **do**{
2. //code to be executed
3. }**while**(**true**);

### C# Infinitive do-while Loop Example

1. **using** System;
2. **public** **class** WhileExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
7. **do**{
8. Console.WriteLine("Infinitive do-while Loop");
9. } **while**(**true**);
10. }
11. }

Output:

Infinitive do-while Loop

Infinitive do-while Loop

Infinitive do-while Loop

Infinitive do-while Loop

Infinitive do-while Loop

ctrl+c

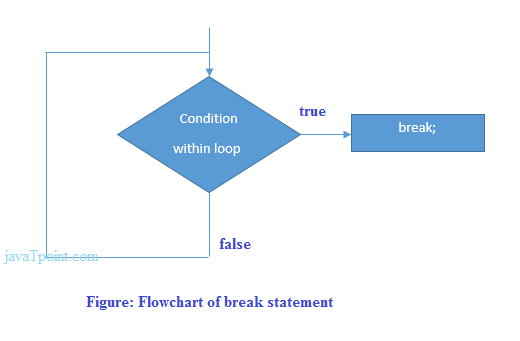
# C# Break Statement

The C# break is used to break loop or switch statement. It breaks the current flow of the program at the given condition. In case of inner loop, it breaks only inner loop.

**Syntax:**

1. jump-statement;
2. **break**;

**Flowchart:**



### C# Break Statement Example

Let's see a simple example of C# break statement which is used inside the loop.

1. **using** System;
2. **public** **class** BreakExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
6. **for** (**int** i = 1; i <= 10; i++)
7. {
8. **if** (i == 5)
9. {
10. **break**;
11. }
12. Console.WriteLine(i);
13. }
14. }
15. }

Output:

1

2

3

4

### C# Break Statement with Inner Loop

The C# break statement breaks inner loop only if you use break statement inside the inner loop. Let's see the example code:

1. **using** System;
2. **public** **class** BreakExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
6. **for**(**int** i=1;i<=3;i++){
7. **for**(**int** j=1;j<=3;j++){
8. **if**(i==2&&j==2){
9. **break**;
10. }
11. Console.WriteLine(i+" "+j);
12. }
13. }
14. }
15. }

Output:

1 1

1 2

1 3

2 1

3 1

3 2

3 3

# C# Continue Statement

The C# continue statement is used to continue loop. It continues the current flow of the program and skips the remaining code at specified condition. In case of inner loop, it continues only inner loop.

**Syntax:**

1. jump-statement;
2. **continue**;

### C# Continue Statement Example

1. **using** System;
2. **public** **class** ContinueExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
6. **for**(**int** i=1;i<=10;i++){
7. **if**(i==5){
8. **continue**;
9. }
10. Console.WriteLine(i);
11. }
12. }
13. }

Output:

2.1M

253

Exception Handling in Java - Javatpoint

1

2

3

4

6

7

8

9

10

### C# Continue Statement with Inner Loop

C# Continue Statement continues inner loop only if you use continue statement inside the inner loop.

1. **using** System;
2. **public** **class** ContinueExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
6. **for**(**int** i=1;i<=3;i++){
7. **for**(**int** j=1;j<=3;j++){
8. **if**(i==2&&j==2){
9. **continue**;
10. }
11. Console.WriteLine(i+" "+j);
12. }
13. }
14. }
15. }

Output:

1 1

1 2

1 3

2 1

2 3

3 1

3 2

3 3

# C# Goto Statement

The C# goto statement is also known jump statement. It is used to transfer control to the other part of the program. It unconditionally jumps to the specified label.

It can be used to transfer control from deeply nested loop or switch case label.

Currently, it is avoided to use goto statement in C# because it makes the program complex.

2.1M

253

Exception Handling in Java - Javatpoint

### C# Goto Statement Example

Let's see the simple example of goto statement in C#.

1. using System;
2. **public** **class** GotoExample
3. {
4. **public** **static** **void** Main(string[] args)
5. {
6. ineligible:
7. Console.WriteLine("You are not eligible to vote!");
9. Console.WriteLine("Enter your age:\n");
10. **int** age = Convert.ToInt32(Console.ReadLine());
11. **if** (age < 18){
12. **goto** ineligible;
13. }
14. **else**
15. {
16. Console.WriteLine("You are eligible to vote!");
17. }
18. }
19. }

Output:

You are not eligible to vote!

Enter your age:

11

You are not eligible to vote!

Enter your age:

5

You are not eligible to vote!

Enter your age:

26

You are eligible to vote!

# C# Comments

The C# comments are statements that are not executed by the compiler. The comments in C# programming can be used to provide explanation of the code, variable, method or class. By the help of comments, you can hide the program code also.

There are two types of comments in C#.

* Single Line comment
* Multi Line comment

### C# Single Line Comment

The single line comment starts with // (double slash). Let's see an example of single line comment in C#.

2.1M

253

Exception Handling in Java - Javatpoint

1. **using** System;
2. **public** **class** CommentExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
6. **int** x = 10;//Here, x is a variable
7. Console.WriteLine(x);
8. }
9. }

Output:

10

### C# Multi Line Comment

The C# multi line comment is used to comment multiple lines of code. It is surrounded by slash and asterisk (/\* ..... \*/). Let's see an example of multi line comment in C#.

1. **using** System;
2. **public** **class** CommentExample
3. {
4. **public** **static** **void** Main(**string**[] args)
5. {
6. /\* Let's declare and
7. print variable in C#. \*/
8. **int** x=20;
9. Console.WriteLine(x);
10. }
11. }

Output:

20