**LOOPS**

In computer programming, a loop is a sequence of [instruction](http://searchcio-midmarket.techtarget.com/definition/instruction)’s that is continually repeated until a certain condition is reached.

OR

A loop statement allows us to execute a statement or a group of statements multiple times.

Real world example-

Suppose a student is giving the exam of 12th class and if he gets fail he/she will have to reattempt exam until he/she gets passed in exam.

Following is the general form of a loop statement in most of the programming languages –



Types of loops :-

* For loop
* Foreach loop
* While loop
* Do While loop
* Nested loop
* Infinite loop

**For Loop**

The 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.

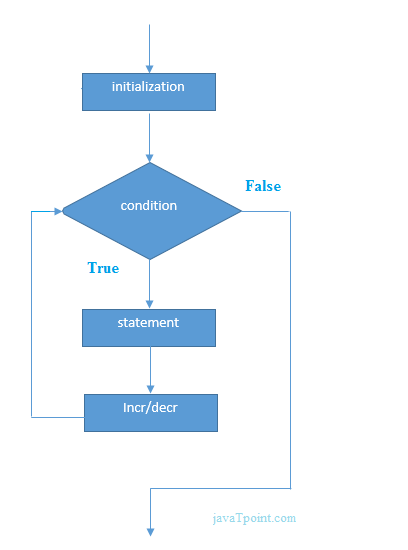
Syntax:

for (initialization; condition; increment/decrement){

//code to be executed

}

**Flowchart:**



**Foreach Loop**

The foreach loop when working with arrays and collections to iterate the items of arrays/collections.

It is used only when there is reading operation perform, means used only for fetching of data not for writing.

Syntax:

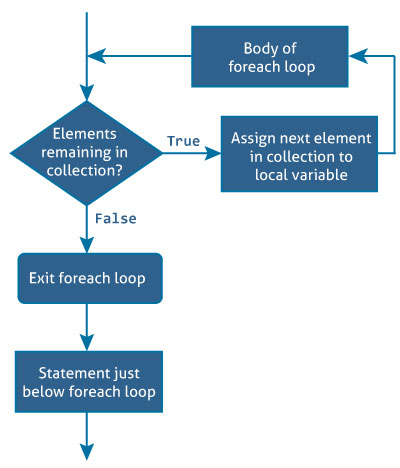
foreach (element in iterable-item)

{

// body of foreach loop

}

Flowchart:



**While Loop**

It repeats a statement or a group of statements while a given condition is true. It tests the condition before executing the loop body.

Syntax:

while(condition) {

//code to be executed

}

Flowchart:



**Do While Loop**

It is similar to a while statement, except that it tests the condition at the end of the loop body, means that do-while loop is executed at least once either condition is true or false.

Syntax:

do {

//code to be executed

} while(condition);

Flowchart:



**Nested Loop**

The syntax for a **nested for loop** -

for (initialization; condition; increment/decrement) {

for (initialization; condition; increment/decrement) {

Statement(s);

}

Statement(s);

}

The syntax for a **nested while loop -**

while (condition) {

while (condition) {

Statement(s);

}

Statement(s);

}

The syntax for a **nested do while loop –**

do {

Statement(s);

do {

Statement(s);

} while (condition);

} while (condition);

**Infinite Loop**

The syntax for an **infinite for loop –**

for ( ;  ; ){

//code to be executed

}

The syntax for an **infinite while loop –**

while(true) {

//code to be executed

}

The syntax for an **infinite do while loop –**

do {

//code to be executed

} while(true);

**Loop Control Statements**

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

Various control statements-

* Break
* Continue
* Goto
* Switch

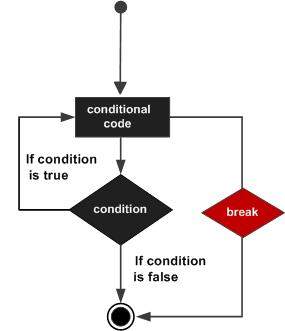
**Break Statement**

The 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:

**break;**

Flowchart:



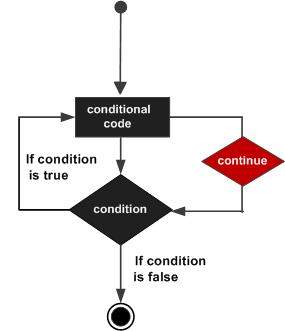
**Continue Statement**

The 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:

**continue;**

Flowchart:



**Goto Statement**

The 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.

**Switch Statement**

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

Syntax:

**switch(expression){**

**case value1: //code to be executed;**

**break;**

**case value2: //code to be executed;**

**break;**

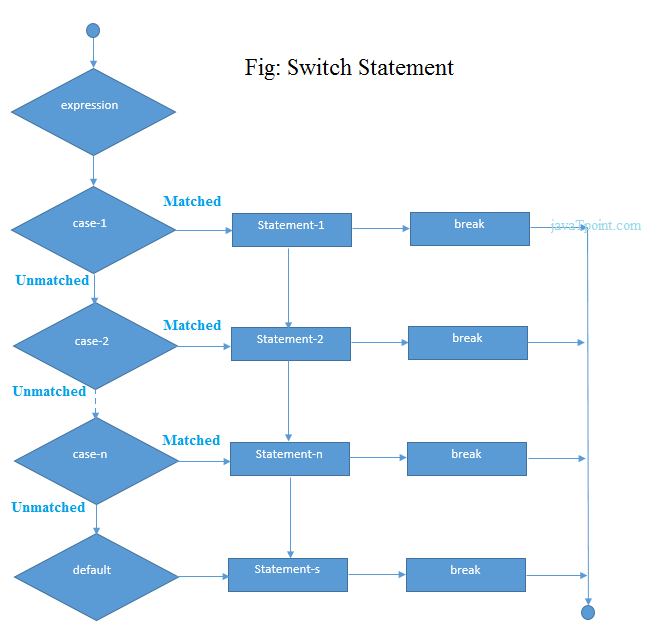
**......**

**default: //code to be executed if all cases are not matched;**

**break;**

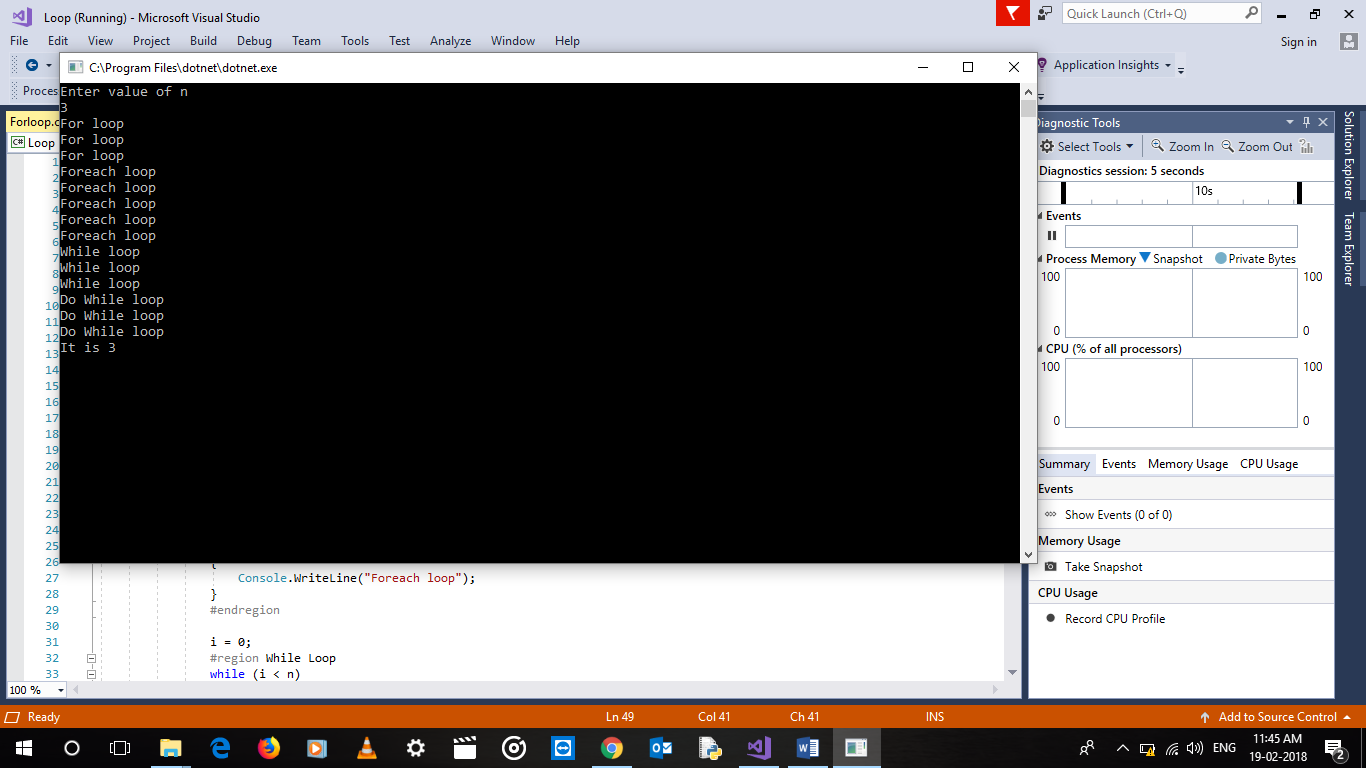
**}**

**Flowchart:**



**Code Snippet**

[**..\source\repos\Loop\Loop\Forloop.cs**](../source/repos/Loop/Loop/Forloop.cs)



**Nested For loop with break and continue statement**

[**..\source\repos\Loop\Loop\Program.cs**](../source/repos/Loop/Loop/Program.cs)

