

Nested for loop

Sometimes, you are required to perform a task in **nested loop** condition.

The **nested loops** are programming constructs consisting of **several loops located into each other**.

A **loop within a loop** is called **nested loop**.

The **innermost** loop is executed **more** times, and the **outermost** — **less** times

The syntax for a **nested while loop** statement in C# is as follows:

```
while(condition)
{
    while(condition)
    {
        statement(s);
    }
    statement(s);
}
```

The syntax for a **nested do...while loop** statement in C# is as follows:

```
do
{
    statement(s);
    do
    {
        statement(s);
    }while( condition );
}while( condition );
```

The syntax for a **nested for loop** statement in C# is as follows:

```

for ( init; condition; increment )
{
    for ( init; condition; increment )
    {
        statement(s);
    }
    statement(s);
}

```

- After **initialization** of the **first for loop**, the **execution** of its body will **start**, which contains the **second (nested) loop**.
- **Second (nested) loop variable** will be **initialized**, its **condition** will be **checked** and the code within its body will be **executed**, then the variable will be updated and execution will **continue** until the condition returns **false**.
- After that the second iteration of the **first for loop will continue**, its **variable will be updated** and the whole second loop will be performed **once again**.
- The inner loop (**second loop**) will be fully **executed** as many times as the body of the outer loop (**first loop**).

Sample Program

```

. using System;
. using System.Collections.Generic;
. using System.Linq;
. using System.Text;
.
. namespace nested_loop
. {
.     class Program
.     {
.         static void Main(string[] args)
.         {
.             int i, j;
.             for (i = 1; i <= 5; i++)
.             {

```

```
.      for (j = 1; j <= i; j++) //Nested for loop
.      {
.          Console.Write(j);
.      }
.      Console.WriteLine("\n");
.      }
.      Console.ReadLine();
.      }
.      }
.      }
```

Output

```
1
12
123
1234
12345
—
```