for loop in C++ with examples

what loop in C++

when we want to execute a statement or set of statements a certain number or repeat the same statement multiple times in the Program. in that case, we use the loop.

In the loop, a statement or set of statements is repeated until the given condition becomes false where a repetition of the loop depends on the loop-size which is user-defined and repetition statement is called **body of loop.**

before going to loop let's know why they are needed.

needs of a loop in C++

sometimes it's happened that we want to execute the same statement/functions multiple times, so in this case, we have two ways to perform these task,

Let's try this with an example

for example, we have to write a statement 5 times,

```
C++ Tutorials // 1.statement executed

C++  Tutorials // 2.statement executed

C++ Tutorials // 3.statement executed

C++ Tutorials // 4.statement executed

C++ Tutorials // 5.statement executed
```

it's fine for 1-2 times but when writing the same statements 5 times then it will not consider as good practice, so overcome to this situation we use loop here

Type of loop in C++

There are 3 types of looping statements in C++, are as follows:

- for loop
- while loop
- do-while loop.

There are three components of a loop–variable initialization, expression/condition and value Up-dation

for loop in C++

In the **for-loop**, there is first **variable-initialization**, then the **condition is given**. If the condition is true then the **body of loop** is executed then after there is an **increment** in the expression value.

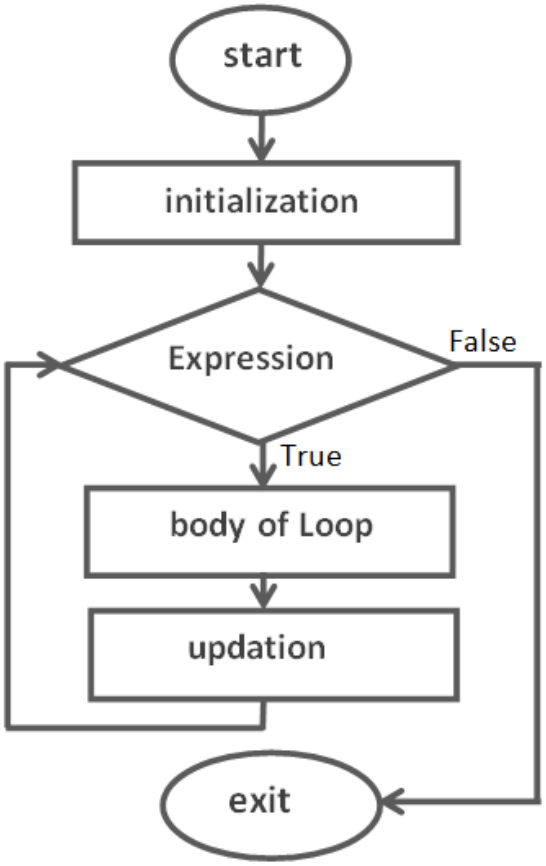
This sequence repeats until the value of the expression is false. As soon as the condition is false, the loop gets terminated and the execution comes out of the loop on the next statement (if available).

SYNTEX-

```
for( initialization; expression; updation)

   {

      body of loop;
```

```
   }
```

in this, **body of loop** may be functions or statements types.



Here is the program,

Example of for-loop in C++

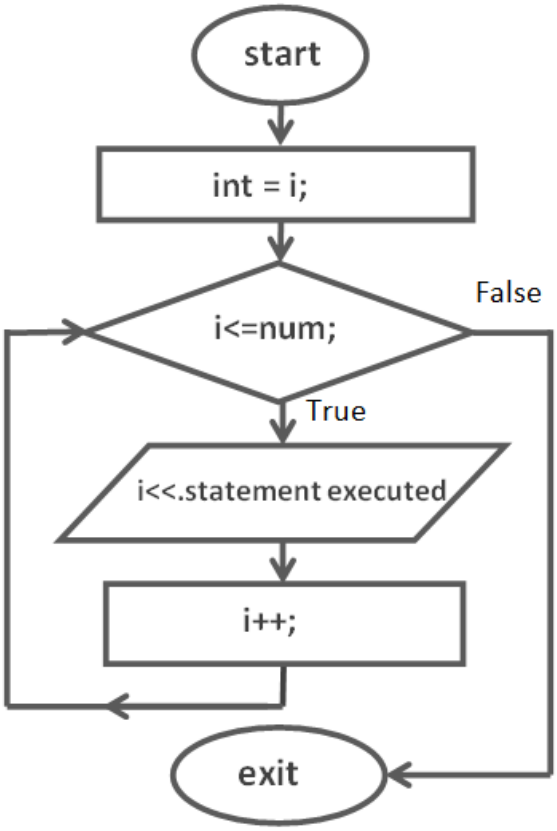
```
#include<iostream>

using namespace std;



 int main()
```

```
  {

     int num = 5;

     for(int i=1; i<=num; i++)

      {

        cout<<i<<".statement executed\n"; // body of loop

      }

return 0;

 }
```

OUTPUT

```
1.statement executed

2.statement executed

3.statement executed

4.statement executed

5.statement executed
```

In this program, for-loop will work as,



Explanation

we give for-loop size 5 by declaring the variable **num = 5** in the program,

now for loop is started, where we have assigned the value of the variable **i = 1**,

After this condition will check, because here the condition becomes true **(1<=5)**, then

body of loop will execute and the following statement will be executed, will execute the following statement,

```
cout<<i<<".statement executed\n";
```

After this, the value of the variable i is increased by 1. Then again the condition will check but now the condition will be (2 <= 5) which is becoming true then the **body of loop** will be executed again.

This sequence repeated until the condition becomes false (6<=5). because here condition here true 5 times, so **body of loop** will be executed 5 times.

Here is the procedure,

when i =1, 1<=5 (True)

```
for(int i=1; 1<=5; 1++)

  {

     1.statement executed

  }
```

the variable i value will be increased by 1 in every statement because of i++ , so i =2, <=5 (True)

```
for(int i=2; 2<=5; 2++)

 {

    2.statement executed

 }
```

i=3, 3<=5 (True)

```
for(int i=3; 3<=5; 3++)

  {

    3.statement executed

  }
```

i=4, 4<=5 (True)

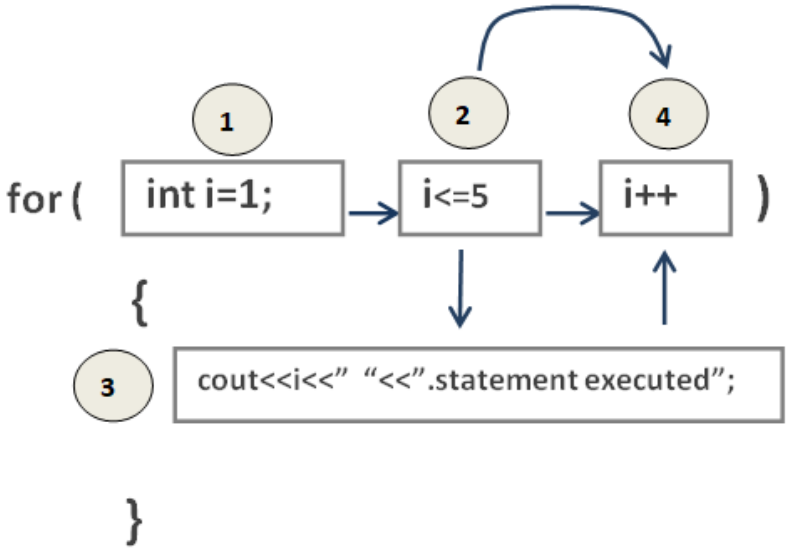
```
for(int i=4; 4<=5; 4++)

  {

    4.statement executed

  }
```

i=5, 5<=5 (True)

```
for(int i=5; 5<=5; 5++)

  {

     5.statement executed

  }
```

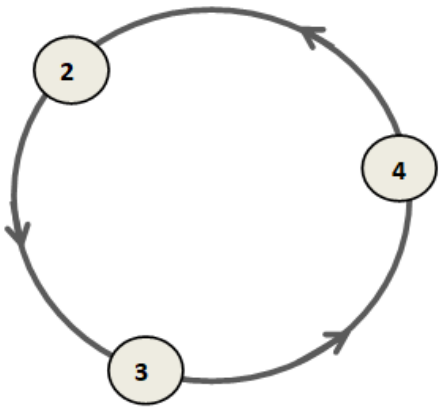
i=6, 6<=5 (False)

```
for(int i=6; 6<=5; 6++)

  {

    statement skip

  }
```



such as, for loop,

number 2, 3 and 4 will repeat until the condition becomes false,



Thus a successful execution of the loop is done.

Here is another way,

The table below shows the implementation of a **for-loop**. Here, every time the initialization is incremented in value 1, then the **body of loop** is executed.

initialization value (i=0)	condition (i<5)	body of loop execute
1	1<=5 (True)	1.statement executed
2	2<=5 (True)	2.statement executed
3	3<=5 (True)	3.statement executed
4	4<=5 (True)	4.statement executed
5	5<=5 (True)	5.statement executed
6	6<=5 (False)	Exit

Now you understand for loop , let's try with some examples here,

In the below program we performed a decrementing loop,

```
#include<iostream>

using namespace std;



  int main()

   {

     int num=5;
```

```
      for(int i = num; i >= 1; i--)

           cout<<i<<"\t";



     return 0;

  }
```

OUTPUT

```
5  4  3  2  1
```

Print the sum of two number until the condition becomes false,

```
#include<iostream>

using namespace std;



  int main()

   {

     int num1, num2;

     for(int i=0; i<3; i++)

      {



         cout<<"Enter Two number: ";

         cin>>num1>>num2;
```

```
        cout<<"sum: "<<num1+num2;

        cout<<"sub: "<<num1-num2;

        return 0;

     }

   cout<<""Program Exiting..;

  }
```

OUTPUT

```
Enter Two number: 4 3

sum: 7

sub: 1



Enter Two number: 2 7

sum: 9

sub: -1



Enter Two number: 9 1

sum: 10

sub: 8
```

```
Program Exiting..
```

start loop with given input and terminate with the user input itself,

```
#include<iostream>

using namespace std;

  int main()

  {

      int start,end;

    cout<<"Enter First Number: ";

    cin>>start;

    cout<<"Enter Last Number: ";

    cin>>end;

     for(int count=start; count<=end; count++)

          cout<<". "<<count<<"\t";

 return 0;

 }
```

OUTPUT

```
Enter First Number: 2

Enter Last Number:  11
```

```
2 3 4 5 6 7 8 9 10 11
```

add two for-loop in a single statement are possible, where in the first loop ‘s value is incrementing and second is decrementing type.

```
#include<iostream>

using namespace std;

  int main()

  {

       int max = 5, min = 0;

    for(int i=0,j=5; j<min,max>=i; i++,j--)

          cout<<i<<"\t"<<j<<"\n";

  return 0;

 }
```

OUTPUT

```
0        5

1        4

2        3

3        2

4        1
```

```
5        0
```

Example of for-loop with if-statement in C++

print only even number using if-statement inside the **body of loop**

```
#include<iostream>

using namespace std;

  int main()

   {

     int num;

     cout<<"Enter Last number: ";

     cin>>num;

      for(int i = 0; i <= num; i++)

        {

           if(i%2==0)

             cout<<i<<"\t";

         }

   return 0;

  }
```

OUTPUT

```
Enter Last Number: 10
```

```
0  2  4   6   8   10
```

like nested if-else, C++ supports nested for-loop.

nested for-loop in C++

when a for-loop is a body of other for-loop it is called nested for-loop. nested for-loop can have little complex but it will perform a normal for-loop.

syntax

```
for( initialization; expression; updation)

    {

 //outer-loop body

      for( initialization; expression; updation)

         {

          //inner-loop body;

         }

      ..................

    }
```

a for-loop body can have more than one for-loop

Example of nested for loop in C++

here we will consider this with a simple example,

```
#include<iostream>
```

```
using namespace std;

  int main()

   {

      int max = 3, min = 0;

//outer-loop

    for(int i=1; i<=3; i++)

     {

       cout<<"First-loop: "<<i<<"\n";

    //inner-loop

       for(int j=1; 1<=2; j++)

           cout<<" Second-loop: "<<j<<endl;

     }

  return 0;

 }
```

OUTPUT

```
First-loop: 1

 Second-loop: 1

 Second-loop: 2

First-loop: 2
```

```
 Second-loop: 1

 Second-loop: 2

First-loop: 3

 Second-loop: 1

 Second-loop: 2
```

Explanation

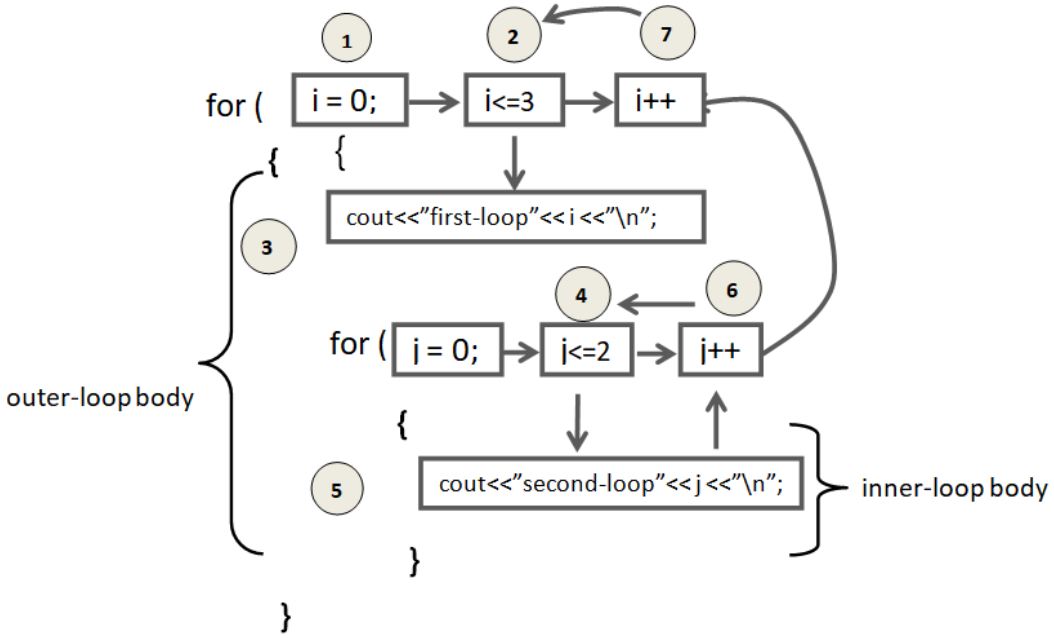
Remember here **outer-loop** size is 3 then it will execute 3 times,

```
First-loop: 1

First-loop: 2

First-loop: 3
```

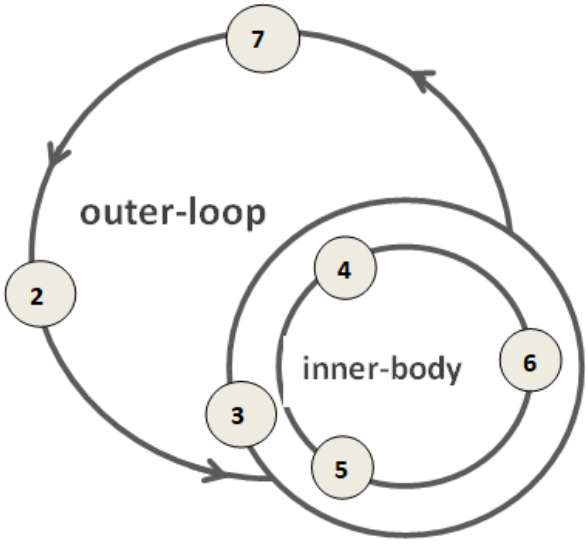
while the **inner-loop** size is 2 so it will execute 2 times,

```
Second-loop: 1

Second-loop: 2
```

here nested loop will execution will as follows diagram,



above codes will execute as



creating patterns programs in C++ is a good example of nested for-loop.

Infinitive loop in C++

In all the above programs, we read when a condition becomes false loop terminates and execution go to the next statement. but what will happen if the condition will never False,

When the loop condition is never false, in this case, the loop will never terminate and it will continue to execute indefinitely,

Example of the infinitive loop in C++

In the below Program condition given is **i<num**, where num = 5; here loop will execute until variable **i** value does not greater than variable **num** which is never happening here because the variable **i** value always decremented [using **i- -]** so in this case, condition always becoming True **(i < 5)**, so this loop will be executed indefinitely.

```
#include<iostream>

using namespace std;

   int main()

  {

    int num = 5;

     for(int i=1; i<=num; i--)

      {

        cout<<i<<".statement executed\n"; // body of loop

      }



 return 0;

}
```

OUTPUT

```
1.statement executed

2.statement executed

3.statement executed

execution will continue....
```

such as,

Related Exercise

- Print natural number until 100 and also their sum
- Print odd number in a descending order till number 100.
- skipping a number at an interval of 10 in a series
- Print alphabet A-Z using for loop
- Execute and Terminate a loop manually by Given numbers
- Calculating Total of even and odd number
- Generate a Table of given number using for loop