# **Lecture Notes No. (7-A):** Loops

Loops are among the most basic and powerful of programming concepts. You may encounter situations, when a block of code needs to be executed several times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on. Programming languages provide various control structures that allow for more complicated execution paths. A loop statement allows us to execute a statement or group of statements multiple times. A loop in a computer program is an instruction that repeats until a specified condition is reached.

There are three looping structures in C++, all of which are used to execute a specific code block multiple time. Just as with the conditional if statement, the curly brackets for the loops can be left out if there is only one statement in the code block.

Depending upon the position of a control statement in a program, looping in C is classified into two types:

1. **Entry-controlled loop:**
   In C++, **while** and **for** are entry-controlled loops.
2. **Exit-controlled loop:**
   In C++, **do while** is exit-controlled loop.

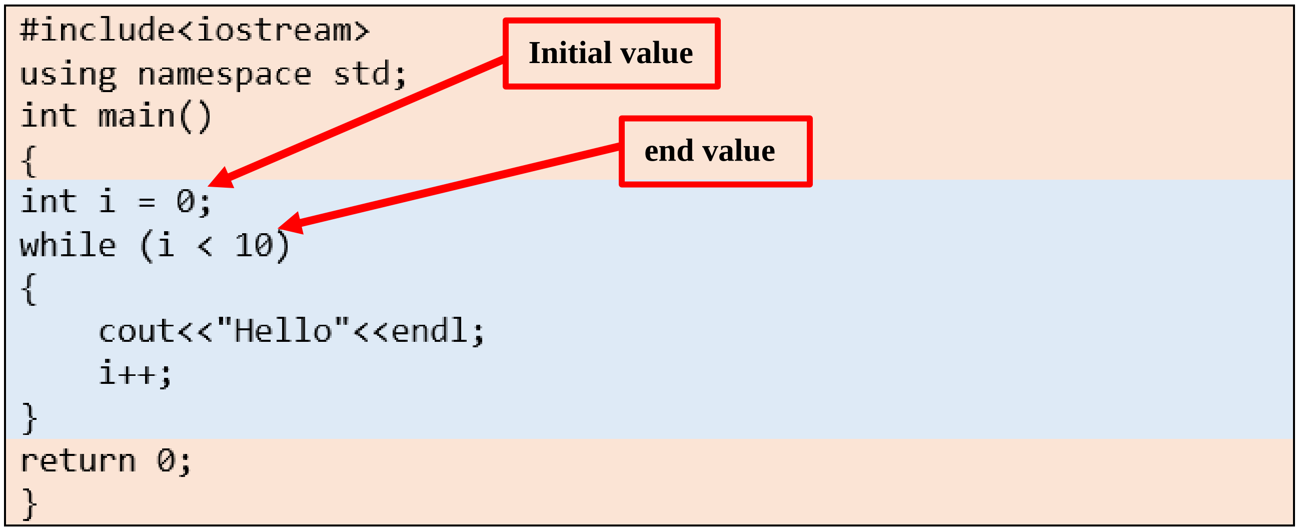
## while

The while loop runs through the code block only if its condition is true, and will continue looping for as long as the condition remains **true**. Bear in mind that the condition is only checked at the start of each iteration.

In the following example, C++ program prints the **Hello** ten times. The program uses a variable **i** as counter. **i** counts how many times the word **Hello** is printed. It starts with value of **0** and is increased by **1** after every time the program will print **Hello**. **while** condition is checked in each iteration. In each iteration, the whole code between the **while** curly brackets is executed until **i** will be equal to 10. Then the condition **i<10** will become **false**.

```cpp
#include<iostream>
using namespace std;
int main()
{
int i = 0;
while (i < 10)
{
    cout<<"Hello"<<endl;
    i++;
}
return 0;
}
```

Programs with counter is one of uses of loops but not the only one. Counter has a start value (initial value) and its value changes until it reaches the end value. Initial value can be greater or smaller than the end value. Therefore, the changes will be in ascending or descending order. The amount of change (increasing or decreasing) is consistent and it can be one or more.



```cpp
#include<iostream>
using namespace std;
int main()
{
int i = 0;
while (i < 10)
{
    cout<<"Hello"<<endl;
    i++;
}
return 0;
}
```

## Exercises (Part1):

1. Write C++ Program to print the numbers between 1 and 10.
2. Write C++ Program to print the numbers between 1 and 100.
3. Write C++ Program to print the numbers between 300 and 320.
4. Write C++ Program to print the numbers between -10 and 10.
5. Write C++ Program to print the numbers between 10 and 1.

6. Write C++ Program to print the numbers between 10 and -10.
7. Write C++ Program to print the even numbers between 1 and 100.
8. Write C++ Program to print the odd numbers between 1 and 100.
9. Write C++ Program to print the numbers between **A** and **B**. Notice that **A** can be smaller or greater than **B**.

## Exercises (Part1) Solutions:

### 1.

```
#include<iostream>
using namespace std;
int main()
{
int i = 1;
while (i <= 10)
{
    cout<< i <<endl;
    i++;
}
return 0;
}
```

### 2.

```
#include<iostream>
using namespace std;
int main()
{
int i = 1;
while (i <= 100)
{
    cout<< i <<endl;
    i++;
}
return 0;
}
```

**3.**

```
#include<iostream>
using namespace std;
int main()
{
int i = 300;
while (i <= 320)
{
    cout<< i <<endl;
    i++;
}
return 0;
}
```

**4.**

```
#include<iostream>
using namespace std;
int main()
{
int i = -10;
while (i <= 10)
{
    cout<< i <<endl;
    i++;
}
return 0;
}
```

## 5.

When the start value is greater than the end value that means we have to count in descending order. So, we have to decrease **i** value. Also, the condition have to be change to repeat while the value of **i** is greater than or equal the end value.

```
#include<iostream>
using namespace std;
int main()
{
int i = 10;
while (i >= 1)
{
    cout<< i <<endl;
    i--;
}
return 0;
}
```

## 6.

```
#include<iostream>
using namespace std;
int main()
{
int i = 10;
while (i >= -10)
{
    cout<< i <<endl;
    i--;
}
return 0;
}
```

## 7.

**There are two ways to solve this program:**

**A:**

```
#include<iostream>
using namespace std;
int main()
{
int i = 1;
while (i <= 100)
{
     if(i%2==0)
     {
          cout<< i <<endl;
     }
    i++;
}
return 0;
}
```

**B:**

```
#include<iostream>
using namespace std;
int main()
{
int i = 2;
while (i <= 100)
{
    cout<< i <<endl;
    i+=2;
}
return 0;
}
```

## 8.

**There are two ways to solve this program:**

**A:**

```
#include<iostream>
using namespace std;
int main()
{
int i = 1;
while (i <= 100)
{
     if(i%2==1)
     {
          cout<< i <<endl;
     }
    i++;
}
return 0;
}
```

**B:**

```
#include<iostream>
using namespace std;
int main()
{
int i = 1;
while (i <= 100)
{
    cout<< i <<endl;
    i+=2;
}
return 0;
}
```

## 9.

```
#include<iostream>
using namespace std;
int main()
{
int i,A,B;
cout<<"Count starts from:";
cin>>A;
cout<<"Count ends with:";
cin>>B;
i=A;
if(B>A)
{
    while (i <= B)
    {
        cout<< i <<endl;
        i++;
    }
}
else
{
    while (i >= B)
    {
        cout<< i <<endl;
        i--;
    }
}
return 0;
}
```

## for

The **for** loop is used to run through a code block a specific number of times. It uses three parameters.

- The first one initializes a counter and is always executed once before the loop.
- The second parameter holds the condition for the loop and is checked before each iteration.
- The third parameter contains the increment (or decrement) of the counter and is executed at the end of each iteration.

**Example**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=0 ; i<10 ; i++)
     {
          cout<<"Hello"<<endl;
     }
return 0;
}
```

## Solve the exercises above (Part1) using for loop:

### 1.

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1 ; i<=10 ; i++)
     {
          cout<<i<<endl;
     }
return 0;
}
```

**2.**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1 ; i<=100 ; i++)
     {
          cout<<i<<endl;
     }
return 0;
}
```

**3.**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=300 ; i<=320 ; i++)
     {
          cout<<i<<endl;
     }
return 0;
}
```

**4.**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=-10 ; i<=10 ; i++)
     {
          cout<<i<<endl;
     }
return 0;
}
```

**5.**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=10 ; i>=1 ; i--)
    {
        cout<<i<<endl;
    }
return 0;
}
```

**6.**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=10 ; i>=-10 ; i--)
    {
        cout<<i<<endl;
    }
return 0;
}
```

## 7.

**There are two ways to solve this program:**

**A:**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1 ; i<=100 ; i++)
     {
     if(i%2==0)
          {
               cout<<i<<endl;
          }
     }
return 0;
}
```

**B:**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=2 ; i<=100 ; i+=2)
     {
         cout<<i<<endl;
     }
return 0;
}
```

## 8.

**There are two ways to solve this program:**

**A:**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1 ; i<=100 ; i++)
     {
     if(i%2==1)
          {
               cout<<i<<endl;
          }
     }
return 0;
}
```

**B:**

```
#include<iostream>
using namespace std;
int main()
{
for(int i=1 ; i<=100 ; i+=2)
     {
          cout<<i<<endl;
     }
return 0;
}
```

## 9.

```
#include<iostream>

using namespace std;

int main()

{

int A,B;

cout<<"Count starts from:";

cin>>A;

cout<<"Count ends with:";

cin>>B;

i=A;

if(B>A)

{

    for(int i=A; i<=B;i++)

    {

        cout<< i <<endl;

    }

}

else

{

    for(int i=A; i>=B;i--)

    {

        cout<< i <<endl;

    }

}

return 0;

}
```

In all previous exercises, the program uses a variable as counter (commonly **i**). The counter counts numbers with consistent increment or decrement. Also, each program is written twice using **while** and **for**, so you can be familiar with each structure. However, using **for** loop is more common in counter situations.

In the following exercises, we will use, in addition to a counter variable, a cumulative storage variable to combine the partial result from each iteration in the loop in order to get the complete result.

## Exercises (Part2):

1. Write C++ Program to find the summation of the numbers between 20 and 50.
2. Write C++ Program to find the summation of the numbers between 35 and 100.
3. Write C++ Program to find the summation of the numbers between A and B.
4. Write C++ Program to find the factorial of 6.
5. Write C++ Program to find the factorial of N, where N is a natural number.
6. Write C++ Program to find the value of x in the following equation:

$$y = x + x^2 + x^3 + \ldots + x^n$$
$$\textit{where n is a positive integer}$$

7. Write C++ Program to find the value of x in the following equation:

$$y = 1 + x + x^2 + x^3 + \ldots + x^n$$
$$\textit{where n is a positive integer}$$

8. Write C++ Program to find the value of x in the following equation:

$$y = x^2 + x^4 + x^6 + \ldots + x^n$$
$$\textit{where n is an even positive integer}$$

9. Write C++ Program to find the value of x in the following equation:

$$y = x^1 + x^3 + x^5 + \ldots + x^n$$
$$\textit{where n is an odd positive integer}$$

10.Write C++ Program to find the value of x in the following equation:

$$y = \frac{x}{2} + \frac{x^2}{4} + \frac{x^3}{6} + \ldots + \frac{x^n}{2n}$$
$$\textit{where n is a positive integer}$$

11.Write C++ Program to find the value of x in the following equation:

$$y = x + \frac{x^2}{3} + \frac{x^3}{5} + \ldots + \frac{x^n}{2n-1}$$
$$\textit{where n is a positive integer}$$

## Exercises (Part2) Solutions:

**1.**

using **for:**

```
#include<iostream>
using namespace std;
int main()
{
int sum=0;
for(int i=20 ; i<=50 ; i++)
     {
          sum +=i;
     }
cout<<<<"The Summation of numbers 20 to 50 is "<<sum;
return 0;
}
```

using **while:**

```
#include<iostream>
using namespace std;
int main()
{
int sum=0,i=20;
while(i<=50)
     {
          sum +=i++;
     }
cout<<<<"The Summation of numbers 20 to 50 is "<<sum;
return 0;
}
```

## 2.

using **for:**

```
#include<iostream>
using namespace std;
int main()
{
int sum=0;
for(int i=35 ; i<=100 ; i++)
     {
          sum +=i;
     }
cout<<<<"The Summation of numbers 35 to 100 is "<<sum;
return 0;
}
```

using **while:**

```
#include<iostream>
using namespace std;
int main()
{
int sum=0,i=35;
while(i<=100)
     {
          sum +=i++;
     }
cout<<<<"The Summation of numbers 35 to 100 is "<<sum;

return 0;
}
```

## 3.

using **for:**

```
#include<iostream>
using namespace std;
int main()
{
int sum=0, A, B;
cout<<"Count starts from: ";

cin>>A;

cout<<"Count ends with: ";

cin>>B;

i=A;

if(B>A)

{

for(int i=A ; i<=B ; i++)
      {
            sum +=i;
      }
}
else
{
for(int i=A ; i>=B ; i--)
      {
            sum +=i;
      }
}
cout<<<<"The Summation of numbers "<< A <<" to "<< B <<" is "<<sum;

return 0;
}
```

using **while:**

```
#include<iostream>
using namespace std;
int main()
{
int sum=0,i,A,B;
cout<<"Count starts from: ";

cin>>A;

cout<<"Count ends with: ";

cin>>B;

i=A;

if(B>A)

{

while(i<=B)
      {
            sum +=i++;
      }
}
else
{
while(i>=B)
      {
            sum +=i--;
      }
}
cout<<<<"The Summation of numbers "<< A <<" to "<< B <<" is "<<sum;

return 0;
}
```

## 4.

using **for:**

```
#include<iostream>
using namespace std;
int main()
{
int fact=1;
for(int i=1 ; i<=6 ; i++)
     {
          fact *=i;
     }
cout<<<<"The factorial of 6 is "<<fact;

return 0;
}
```

using **while:**

```
#include<iostream>
using namespace std;
int main()
{
int fact=1 , i=1;
while(i<=6)
     {
          fact *=i++;
     }
cout<<<<"The factorial of 6 is "<<fact;

return 0;
}
```

**5.**

using **for:**

```
#include<iostream>
using namespace std;
int main()
{
int fact=1,N;
cout<<"Enter a natural number to find its factorial: ";

cin>>N;

for(int i=1 ; i<=N ; i++)
     {
          fact *=i;
     }
cout<<<<"The factorial of "<< N <<" is "<<fact;

return 0;
}
```

using **while:**

```
#include<iostream>
using namespace std;
int main()
{
int fact=1, i=1, N;
cout<<"Enter a natural number to find its factorial: ";

cin>>N;

while(i<=N)
     {
          fact *=i++;
     }
cout<<<<"The factorial of "<< N <<" is "<<fact;

return 0;
}
```

In the following exercises, mathematical functions will be used with the loop counters to find the value of the series. **for** is used to solve the rest of the exercises. Try to solve them using **while** yourself.

**٦.**

```
#include<iostream>

#include<cmath>

using namespace std;

int main()

{

int n, x, y=0;

cout<<"Enter the value of n in the series, n= ";

cin>>n;

cout<<"x= ";

cin>>x;

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

{

 y +=pow(x,i);

}

cout<<"y="<<y;

return 0;

}
```

**7.**

```
#include<iostream>

#include<cmath>

using namespace std;
```

```
int main()
{
int n, x, y=0;
cout<<"Enter the value of n in the series, n= ";
cin>>n;
cout<<"x= ";
cin>>x;
for(int i=0; i<=n;i++)
{
 y +=pow(x,i);
}
cout<<"y="<<y;
return 0;
}
```

**8.**

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
{
int n, x, y=0;
cout<<"Enter the value of n in the series, n= ";
cin>>n;
cout<<"x= ";
cin>>x;
for(int i=2; i<=n;i+=2)
{
 y +=pow(x,i);
}
cout<<"y="<<y;
return 0;
}
```

## 9.

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
{
int n, x, y=0;
cout<<"Enter the value of n in the series, n= ";
cin>>n;
cout<<"x= ";
cin>>x;
for(int i=1; i<=n;i+=2)
{
 y +=pow(x,i);
}
cout<<"y="<<y;
return 0;
}
```

## 10.

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
{
int n;
double y=0.0 ,x;
cout<<"Enter the value of n in the series, n= ";
cin>>n;
cout<<"x= ";
cin>>x;
for(int i=1; i<=n;i++)
{
 y +=pow(x,i)/(2.0 * i);
}
cout<<"y="<<y;
return 0;
}
```

## 11.

```
#include<iostream>
#include<cmath>
using namespace std;
int main()
```

```
{
int n;
double y=0.0 ,x;
cout<<"Enter the value of n in the series, n= ";
cin>>n;
cout<<"x= ";
cin>>x;
for(int i=1; i<=n;i++)
{
 y +=pow(x,i)/(2.0 * i - 1);
}
cout<<"y="<<y;
return 0;
}
```