principles of programming

Loops

Department of Cybersecurity

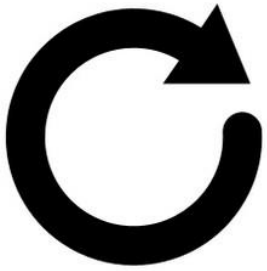
6

Repetitions: Loops

In computer programming, loops are used to repeat a block of code.

A loop is a sequence of instruction s that is continually repeated until a certain condition is reached.

For example, let's say we want to show a message 100 times. Then instead of writing the print statement (cout) 100 times, we can use a loop.



Loops

- C++ provides three types of loops

for loops (1-n times)

  - Repeat a section of code known number of times

while loops (0 –more times)

  - Loop is used to repeat a specific block of code an **unknown** number of times

do while loops (1 –more times)

  - A do while loop is a control flow statement that executes a block of code at least once, and then repeatedly executes the block

Repeat some work

- **Do some repeated work**
  - **Initialization** (Start number)
  - **Condition** (do repeat action until satisfy some condition)
  - **Update** (Next Value)
- **Example**
  - **Initialization** Start with 1
  - **Condition** Count up to 50
  - **Update** Count 1 by 1

For Loop

- **Syntax**

for (*initialization*; *condition*; *update*)

```
{
  statement(s)
  }
```

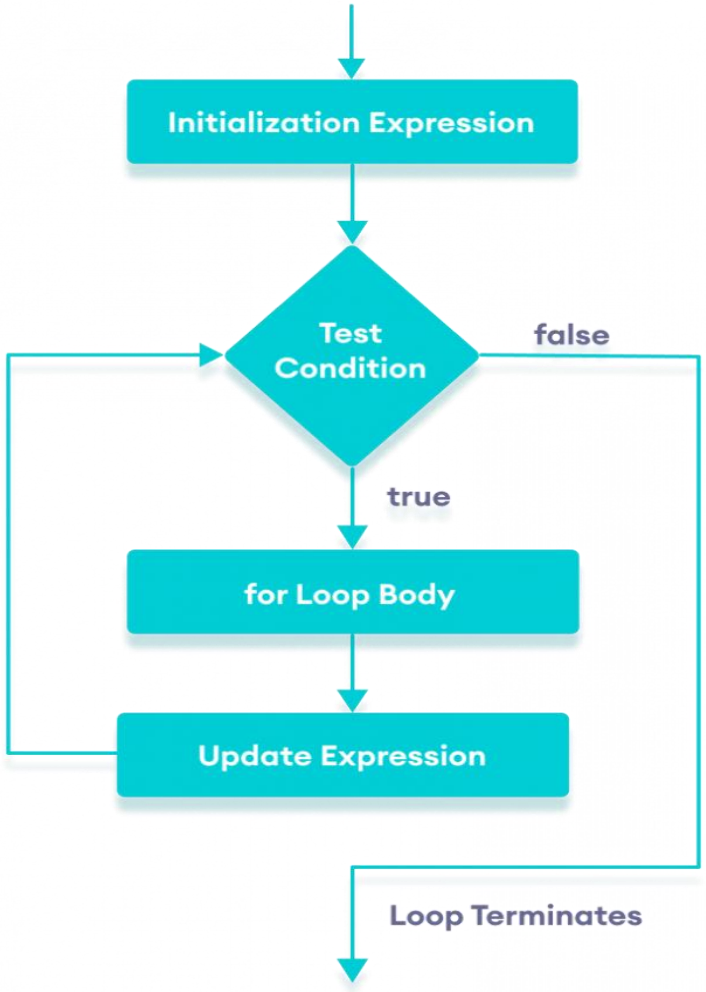
```
for(int i = 1 ; i <11 ; i ++)
{
    cout << "Value "<< i << endl;
}
```

initialization - initializes variables and is executed only once

condition - if true, the body of for loop is executed
if false, the for loop is terminated

update - updates the value of initialized variables and again checks the condition

Flowchart of for Loop in C++



Example : Printing Numbers from 1 to 5

```
#include <iostream>

using namespace std;

int main() {
        for (int i = 1; i <= 5; ++i) {
        cout << i << " ";
    }
    return 0;
}
```

Output

1 2 3 4 5

Example : Display a text 5 times

```
// C++ Program to display a text 5 times

#include <iostream>

using namespace std;

int main() {
    for (int i = 1; i <= 5; ++i) {
        cout <<  "Hello World! " << endl;
    }
    return 0;
}
```

Output

```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

Example : Find the sum of first n Natural Numbers

```
int main() {
    int num, sum;
    sum = 0;

    cout << "Enter a positive integer: ";
    cin >> num;

    for (int i = 1; i <= num; ++i) {
        sum += i;
    }

    cout << "Sum = " << sum << endl;

    return 0;
}
```

Output

Enter a positive integer: 10

Sum = 55

infinite for loop

If the condition in a for loop is always true, it runs forever (until memory is full).

```
// infinite for loop

for(int i = 1; i > 0; i++) {
    // block of code
}
```

In the above program, the condition is always true which will then run the code for infinite times.

Example : Display Multiplication table up to 10

```
int main()
{    int n;
cout << "Enter a positive integer: ";
 cin >> n;
 for (int i = 1; i <= 10; ++i)
{        cout << n << " * " << i << " = " << n * i << endl;
 }

 return 0;}
```

Output

```
Enter an integer: 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Example: Find the Factorial of a Given Number

```
int main() {
 int n;    long factorial = 1.0;
 cout << "Enter a positive integer: ";
 cin >> n;

for(int i = 1; i <= n; ++i)
 {            factorial *= i;
 }
 cout << "Factorial of " << n << " = " << factorial;

 return 0;
}
```

Output

Enter a positive integer: 4
Factorial of 4 = 24

Nested for Loops

A loop can be nested inside of another loop. C++ allows at least 256 levels of nesting.

Syntax

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

Write a C++ program to display the following Multiplication table

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	14	16	18	20	22	24
3	3	6	9	12	15	18	21	24	27	30	33	36
4	4	8	12	16	20	24	28	32	36	40	44	48
5	5	10	15	20	25	30	35	40	45	50	55	60
6	6	12	18	24	30	36	42	48	54	60	66	72
7	7	14	21	28	35	42	49	56	63	70	77	84
8	8	16	24	32	40	48	56	64	72	80	88	96
9	9	18	27	36	45	54	63	72	81	90	99	108
10	10	20	30	40	50	60	70	80	90	100	110	120
11	11	22	33	44	55	66	77	88	99	110	121	132
12	12	24	36	48	60	72	84	96	108	120	132	144