Introduction to Programming
for loop

Muhammad Salman

ABASYN UNIVERSITY

Summary

- Loop
- for loop
- Flow chart
- Syntax
- Programs

Loop

- A statement or set of statement that is executed repeatedly is called loop. The statement(s) in a loop are executed for a specified number of times or until some given condition remain true.
  - In C++ there are three kind of loop statements these are:
    - The while loop
    - The do-while loop
    - The for loop

for Loop

A "For" Loop is used to repeat a specific block of code for a known number of times.

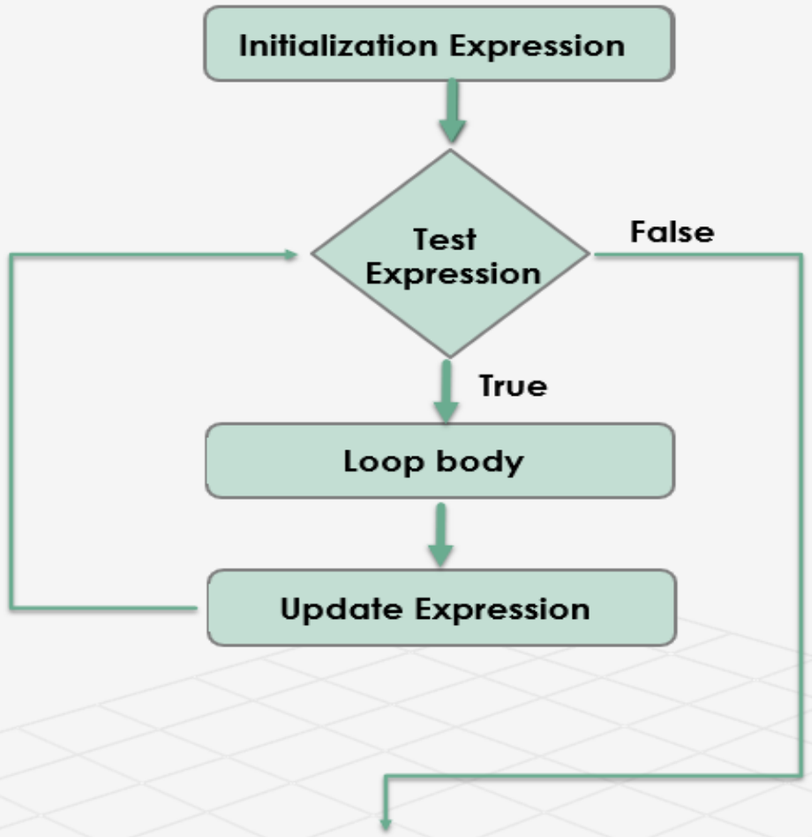
For example,

- if we want to check the grade of every student in the class, we loop from 1 to that number.
- Print the odd numbers from 1 to 1001
- Search a list (array) of numbers for the biggest grade. Again, the computer "knows" how many grades there are, so a for loop is appropriate.

for Loop (Flow chart)

There are three things
in a loop structure i.e. (i) initialization, (ii) a continuation/termination condition and
(iii) changing the value of the condition variable, usually the increment or decrement of the variable
value.

The syntax of for loop is as given below:

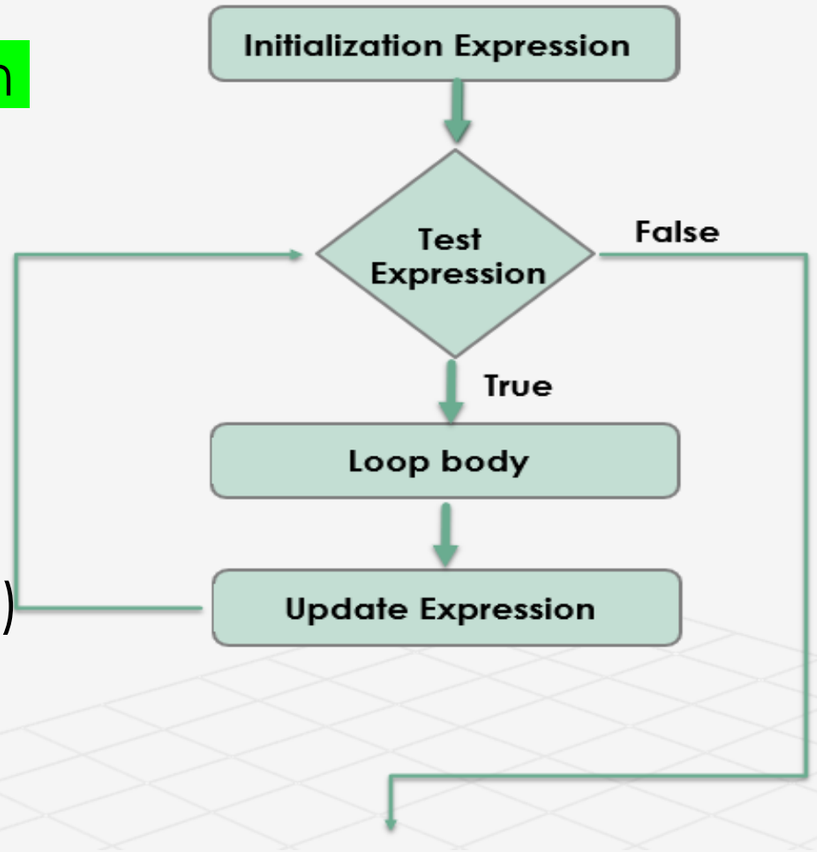



```
for ( initialization; continuation ; incrementing condition )
{
 statment(s)
 }

 for (int i = 0; i < 10; i++)
 cout <<i;
```

for Loop (syntax)

(i) initialization, (ii) a continuation/termination condition
(iii) increment or decrement



"for" is a keyword/ reserved word for; for loop

```
for ( initialization; continuation ; incrementing condition )
{
 statment(s)
 }
```

```
for (int i = 0; i < 10; i++)

cout <<i;
```

```
for (int i = 10; i >=1; i--)

cout <<i;
```

Sample Program

Write a for-loop program to print first ten natural number

```cpp
#include <iostream>
using namespace std;

int main() {
for (int i = 0; i <=10; i++)
cout <<i<< " ";
    return 0;
}
```

```
0 1 2 3 4 5 6 7 8 9 10

=== Code Execution Successful ===
```

Initialization: int i = 0 starts the variable i at 0

continuation/termination condition i <= 10 - continues as long as i is less than or equal to 10

Increment: i++ i=i+1 increases i by 1 after each iteration i=i+1

statment(s): Prints each value of I, i.e., 0 to 10 followed by a space

Sample Program

```cpp
#include <iostream>
using namespace std;

int main() {
for (int i = 10; i >=1; i--)
cout <<i<< " ";
    return 0;
}
```

```
10 9 8 7 6 5 4 3 2 1

=== Code Execution Successful ===
```

Initialization: int i = 10 starts the variable i at 10

continuation/termination condition i >= 1 continues as long as i is greater than or equal to 1

Decrement: i-- i=i-1 decreases i by 1 after each iteration

statment(s): Prints each value of i i.e., 10 to 1 followed by a space

C++ for loop

Write a C++ for loop to calculate the sum of odd number.

```cpp
#include <iostream>
using namespace std;

int main() {
    cout <<"Enter a n to calculate the sum of odd number"<<endl;
    int number ;
    int sum=0;
    cin >>number;
    for (int i = 1; i<=number ; i+=2){
        sum = sum +i;
        cout <<" i is "<<i<<endl;
        //cout <<sum<<endl;
    }
    cout <<"sum =" <<sum;
    return 0;
}
```

```
Enter a n to calculate the sum of odd number
10
 i is 1
 i is 3
 i is 5
 i is 7
 i is 9
sum =25
Process returned 0 (0x0)   execution time :
Press any key to continue.
```

C++ for loop to find factorial of a given number

Write a C++ for loop to find factorial of a given number.

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Enter a number to find factorial: ";
    int number;
    cin >> number;

    int factorial = 1;

    for (int i = 1; i <= number; i++) {
        factorial *= i;
    }

    cout << "Factorial of " << number << " is: " << factorial << endl;
    return 0;
}
```

```
Enter a number to find factorial: 4
Factorial of 4 is: 24

Process returned 0 (0x0)   execution
Press any key to continue.
```

C++ for loop to find factorial of a given number

Write a C++ for loop to find factorial of a given number.

```cpp
#include <iostream>
using namespace std;
int main() {
    cout << "Enter a number to find factorial: ";
    int number;
    cin >> number;
    int factorial = 1;
    // Show the calculation steps (e.g., "4 × 3 × 2 × 1")
    cout << "Calculation: ";
    for (int i = number; i >= 1; i--) {
        cout << i;
        if (i != 1) {
            cout << " x ";
        }
    }
    // Compute the actual factorial
    for (int i = 1; i <= number; i++) {
        factorial *= i;
    }
    cout << " = " << factorial << endl;
    return 0;
}
```

```
Enter a number to find factorial: 4
Calculation: 4 x 3 x 2 x 1 = 24

Process returned 0 (0x0)   execution time : 1.847 s
Press any key to continue.
```

Continue when to use???

The continue statement in a for loop skips the current iteration and jumps to the next one.

It’s useful when you want to:

```
#if not code or (amount == 0 or None):
continue
```

- Skip specific values (e.g., ignore even numbers).
- Avoid executing unnecessary code for certain cases.
- Improve efficiency by bypassing irrelevant iterations.

Continue when to use???

Write a C++ for loop to iterate over the following array and retrieve only the odd numbers.

```
Odd numbers from 1 to 10:
1 3 11 41
Process returned 0 (0x0)   execution time : 0.015 s
Press any key to continue.
```

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Odd numbers from 1 to 10:" <<endl;
int arr[8]= {1,2,3,4,11,6,41,8};
    for (int i : arr) {
        if (i % 2 == 0) {
            continue; // Skip even numbers
        }
        cout << i << " ";
    }
    return 0;
}
```

Continue when to use???

```
Odd numbers from 1 to 10:
1 3 11 41
Process returned 0 (0x0)   execution time : 0.015 s
Press any key to continue.
```

```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Odd numbers from 1 to 10:" << endl;
    int arr[8] = {1, 2, 3, 4, 11, 6, 41, 8};

    for (int i = 0; i < 8; i++) {
        if (arr[i] % 2 != 0) {
            cout << arr[i] << " ";
        }
    }

    return 0;
}
```

The if statement checks if u_name is not equal to name.

Boolean Datatype

- The boolean data type in C++ is used to store true/false values. It's represented by the bool keyword and can only take two possible values:
- true (which is internally represented as 1)
- false (which is internally represented as 0)

```cpp
#include <iostream>

using namespace std;

int main()
{
    bool is_in_islamabad = false;
    bool is_student =false ;
    if (is_in_islamabad && is_student){
        cout <<"Your are living in student, and living in Islamabad "<<endl;
    }
    else if (is_in_islamabad && ! is_student){
        cout <<"You are in Islamabad but you are not student"<<endl;
    }
    else if (!is_in_islamabad && is_student){
        cout <<"you are not in Islamabad but you are student"<<endl;
    }
    else {
        cout <<"You are neither in Islamabad nor a student"<<endl;
    }
    return 0;
}
```

```
C:\Users\Admin\Desktop\boolean\bin\Debug\boolean.exe
You are neither in islamabad nor a student

Process returned 0 (0x0)   execution time : 0.014 s
Press any key to continue.
```

The if statement checks if u_name is not equal to name.

Guessing game.

```cpp
#include <iostream>

using namespace std;

int main() {
    string mySecret = "Moon";
    int guess_count = 0;
    string guess;
    int guess_limit = 3;
    bool out_of_guess = false;

    while (guess != mySecret && !out_of_guess) {
        if (guess_count < guess_limit) {
            cout << "Guess my secret: ";
            cin >> guess;

            guess_count++;
        if (guess_count ==2)
        {
        cout <<"Last warning"<<endl;
       }
        }
         else {
            out_of_guess = true;
        }
    }

    if (out_of_guess) {
        cout << "Nah, it is not !!!" << endl;
    } else {
        cout << "Yeah !!! you got it " << endl;
    }

    return 0; // Success
}
```

C:\Users\Admin\Desktop\guessinggame\main.exe

```
Guess my secret: moon
Guess my secret: star
Last warning
Guess my secret: nature
Nah, it is not !!!

Process returned 0 (0x0)   execution time : 24.513 s
Press any key to continue.
```

C:\Users\Admin\Desktop\guessinggame\main.exe

```
Guess my secret: Moon
Yeah !!! you got it

Process returned 0 (0x0)   execution time : 3.464 s
Press any key to continue.
```

Thanks