Programming Fundamentals with C++

Lecture 9 – Loop Statements

Dr. Muhammad Sajjad

RA: Asad Ullah

Overview

- **Introduction to Loops**
  - What are Loops?
  - Why do we use Loops?
  - Types of Loops in C++
- **The for Loop**
  - What is for loop
  - Syntax of for loop
  - How for loop works
  - Flow Chart of for loop
  - Code Example

Introduction to Loops

What are Loops?

- In Programming, sometimes there is a need to perform some operation **more than once** or (say) **n number** of times.
- Loops come into use when we need to repeatedly execute a block of statements.
- **For example**: Suppose we want to print “Hello World” 10 times. This can be done in two ways as shown below:

Manual Method

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

int main()
{
    cout << "Hello World\n";
    cout << "Hello World\n";
    cout << "Hello World\n";
    cout << "Hello World\n";
    cout << "Hello World\n";
    // write 5 more times
    return 0;
}
```

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

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

Using Loop

Introduction to Loops

Why do we use loops?

- **Efficiency**: Loops save time by automating repetitive tasks.
  - Example: Printing numbers from 1 to 100.
- **Maintainability**: Instead of writing the same code multiple times, we write it once and let the loop handle repetition.

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

int main()
{
    cout << “1\n";
    cout << “2\n";
    cout << “3n";
    cout << “4\n";
    cout << “5\n";
    // 94 lines
    cout<<“100”;
    return 0;
}
```

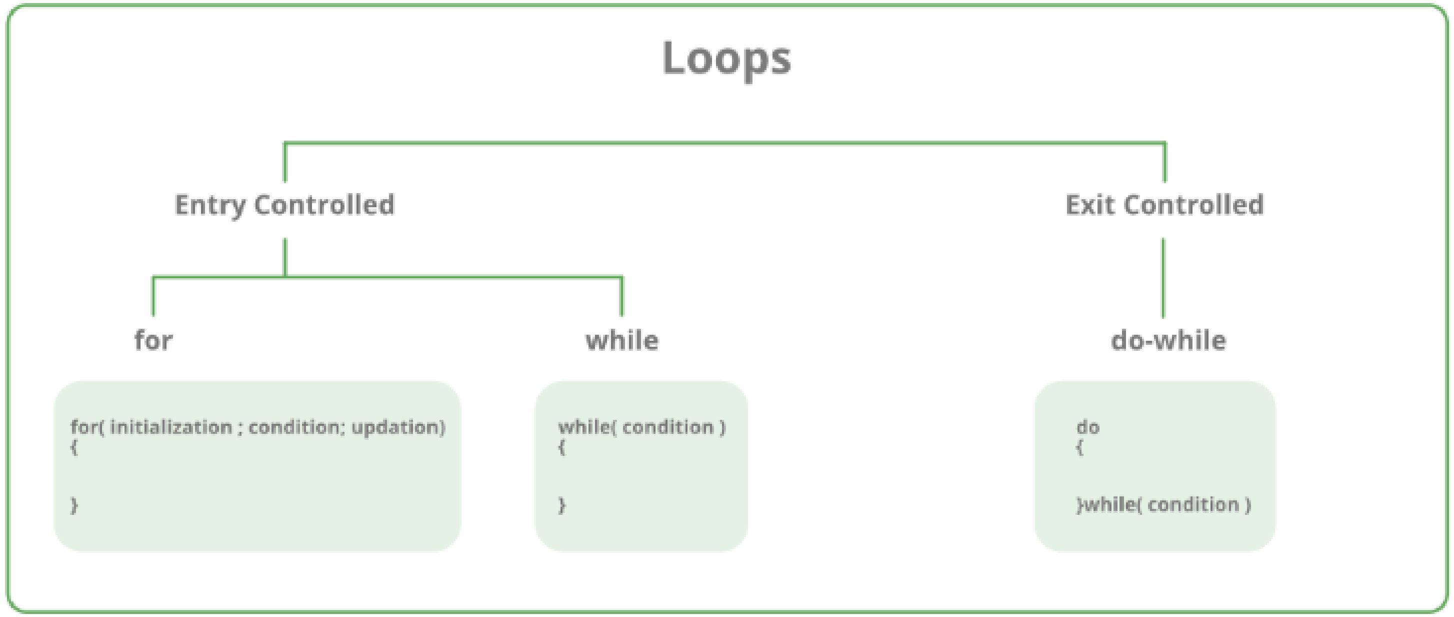
```
#include <iostream>
using namespace std;

int main()
{
    for (int i = 1; i< 101; i++)
    cout << i<<"\n";
     return 0;
}
```

Types of Loops

There are mainly two types of loops:

1. **Entry Controlled loops**: In this type of loop, the test condition is tested before entering the loop body. **For Loop** and **While Loop** is entry-controlled loops.
2. **Exit Controlled Loops**: In this type of loop the test condition is tested or evaluated at the end of the loop body. Therefore, the loop body will execute at least once, irrespective of whether the test condition is true or false. the **do-while loop** is exit controlled loop.



```
for( initialization ; condition; updation)
{

}
```

```
while( condition )
{

}
```

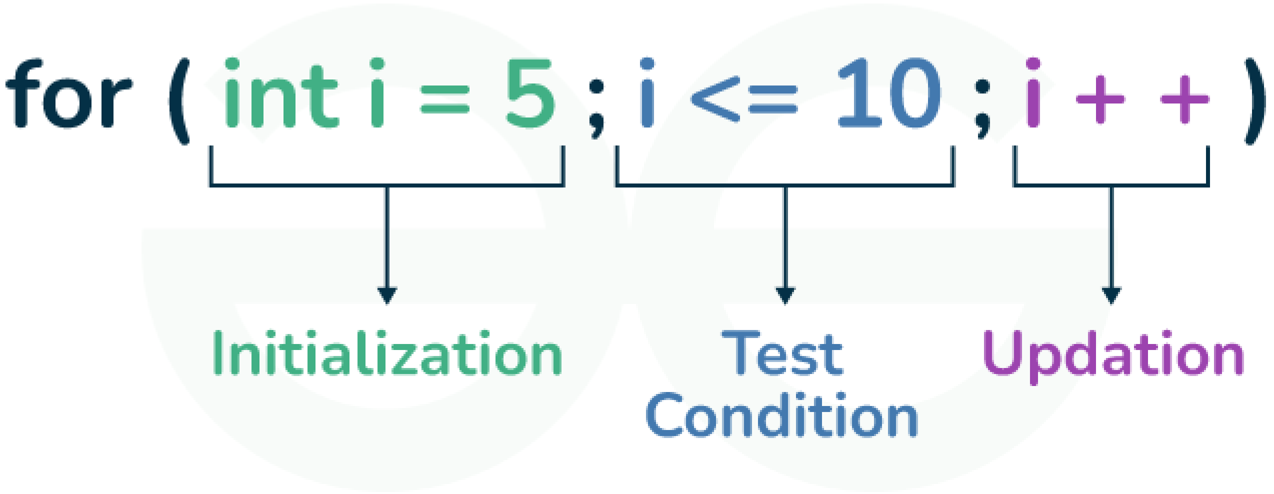
```
do
{

}while( condition )
```

The for Loop

What is for loop

- The for loop is one of the most commonly used loops in programming.
- It is designed for situations where you know in advance how many times you want to repeat a block of code.



```
for ( int i = 5 ; i <= 10 ; i + + )
```

The for Loop

Syntax of for loop

```
for (initialization expr ; condition expr; update expr) {
            //     Body of the loop
           //      Statements we want to execute
}
```

- **Initialization statement:** This statement gets executed only once, at the beginning of the for loop. Variable defined here is only valid in the scope of the loop.
- **Condition:** This statement gets evaluated ahead of each execution of the loop body, and abort the execution if the given condition get false.
- **Iteration execution:** This statement gets executed after the loop body, ahead of the next condition evaluated, unless the for loop is aborted in the body (by break, goto, return or an exception being thrown.)

The for Loop

How for loop works

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

int main()
{
    for (int i = 1; i< 11; i++)
    cout << i<<"\n";
     return 0;
}
```

Step 1: Initialization: The loop variable is set to its initial value.

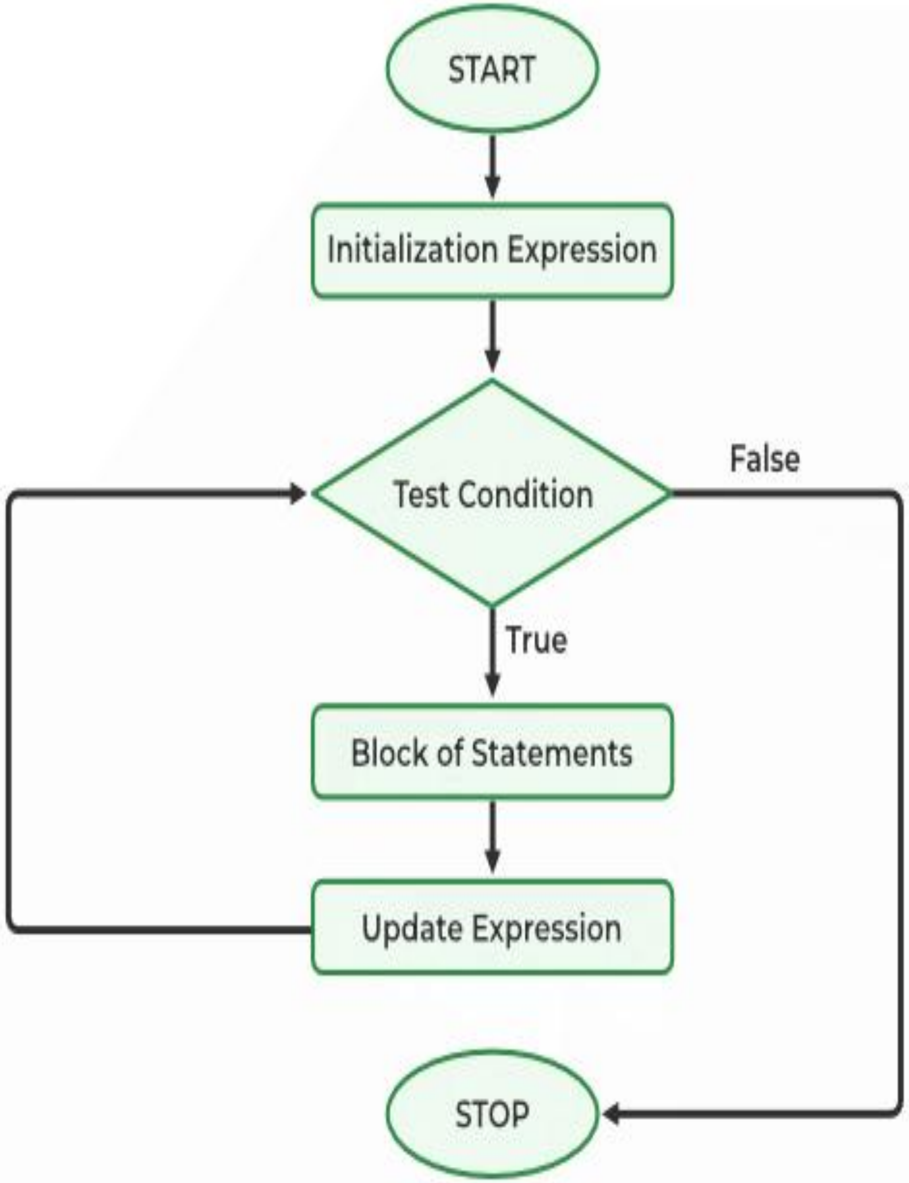
Step 2: Condition Check: The condition is evaluated.

  - If true, the loop continues.
  - If false, the loop stops.

- **Step 3: Code Execution:** The code block inside the loop runs.
- **Step 4: Update:** The loop variable is updated, and the process repeats.

The for Loop

- **Flow Chart of for loop**
  1. Control falls into the for loop. Initialization is done.
  2. The flow jumps to Condition.
  3. Condition is tested.
     - If the Condition yields true, the flow goes into the Body.
     - If the Condition yields false, the flow goes outside the loop.
  4. The statements inside the body of the loop get executed.
  5. The flow goes to the update.
  6. Updating takes place and the flow goes to Step 3 again.
  7. The for loop has ended and the flow has gone outside.



The for Loop

Code Example

Print the multiplication table for 5.

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

int main() {
    int number = 5;
    for (int i = 1; i <= 10; i++) {
        cout << number << " x " << i << " = " << number * i << endl;
    }
    return 0;
}
```

1_Some_Basic_Programs.cpp

2_printing_abc.cpp

3_printing_table_of_a_number.cpp

4_Weekly_Savings_Tracker.cpp

5_Temperature_Conversion_Table.cpp

6_Find_Factorial_Of_Number.cpp

7_Power_Of_Number.cpp

Thank You