Programming Fundamentals with C++

Lecture 10 – Loop Statements

Dr. Muhammad Sajjad

RA: Asad Ullah

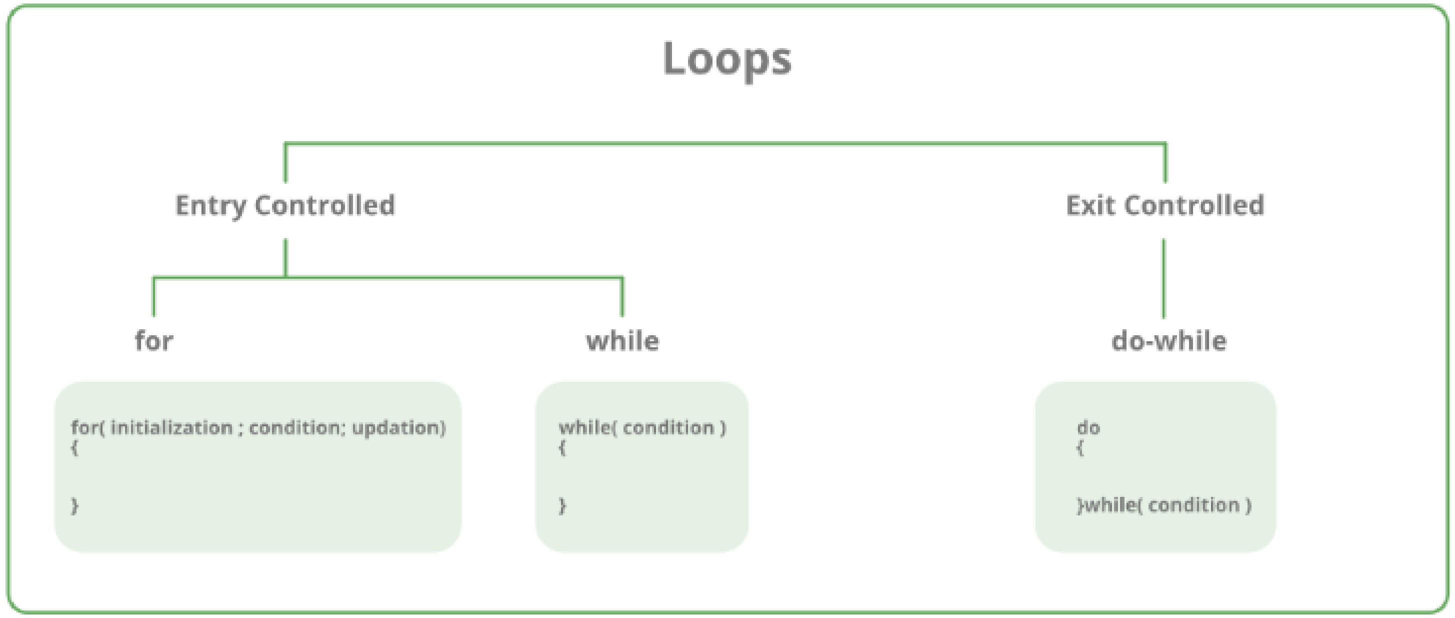
Overview

- **The while Loop**
  - What is while loop
  - Syntax of while loop
  - How while loop works
  - Flow Chart of while loop
  - Code Example

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 )
{

}
```

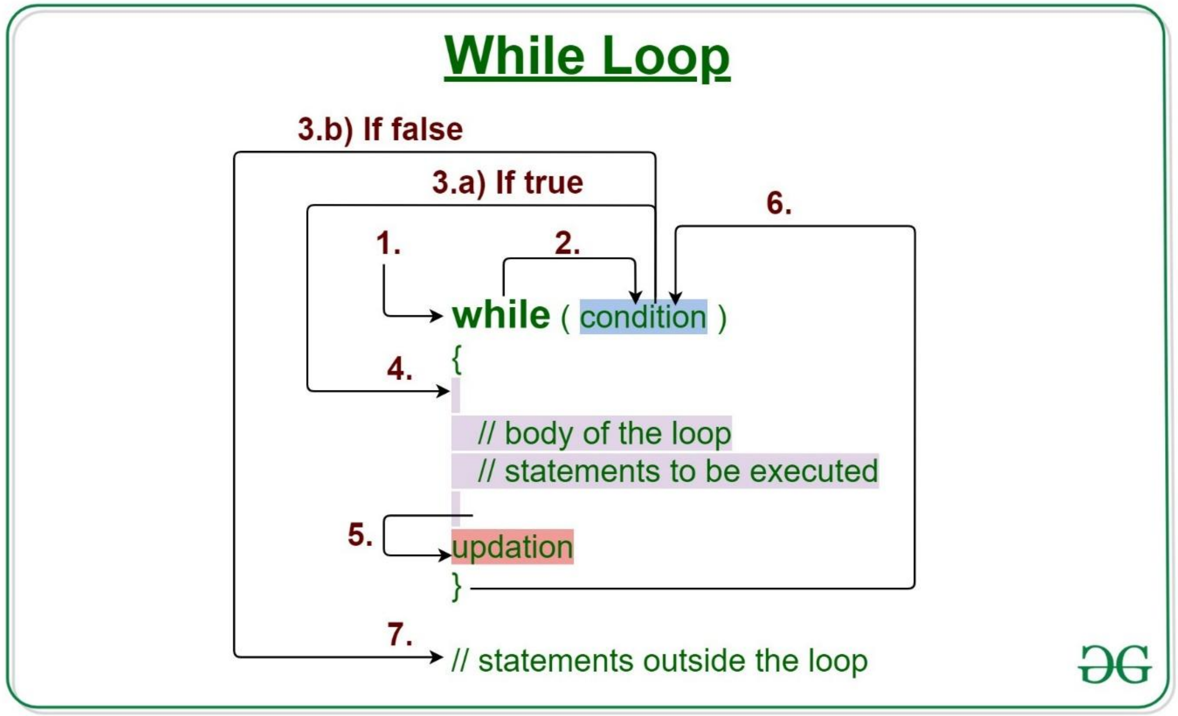
```
do
{

}while( condition )
```

The while Loop

What is while loop

- A while loop is a control structure in programming that allows you to repeatedly execute a block of code as long as a specified condition is true.
  - **Key Idea:** If you want to repeat a task but don't know how many times you need to repeat it, a while loop is ideal. The loop continues until the condition becomes false.



The while Loop

Syntax of while loop

```
initialization expression;
While(test_expression) {
        // statements
        update_expression
}
```

The various **parts of the While loop** are:

- **Test Expression:** In this expression, we have to test the condition. If the condition evaluates to true then we will execute the body of the loop and go to update expression. Otherwise, we will exit from the while loop.
- **Update Expression**: After executing the loop body, this expression increments/decrements the loop variable by some value.
- **Body:** This is a group of statements that include variables, functions, and so on. With the while loop, code, and simple names can be printed, complex algorithms can be executed, or functional operations can be performed.

The while Loop

How while loop works

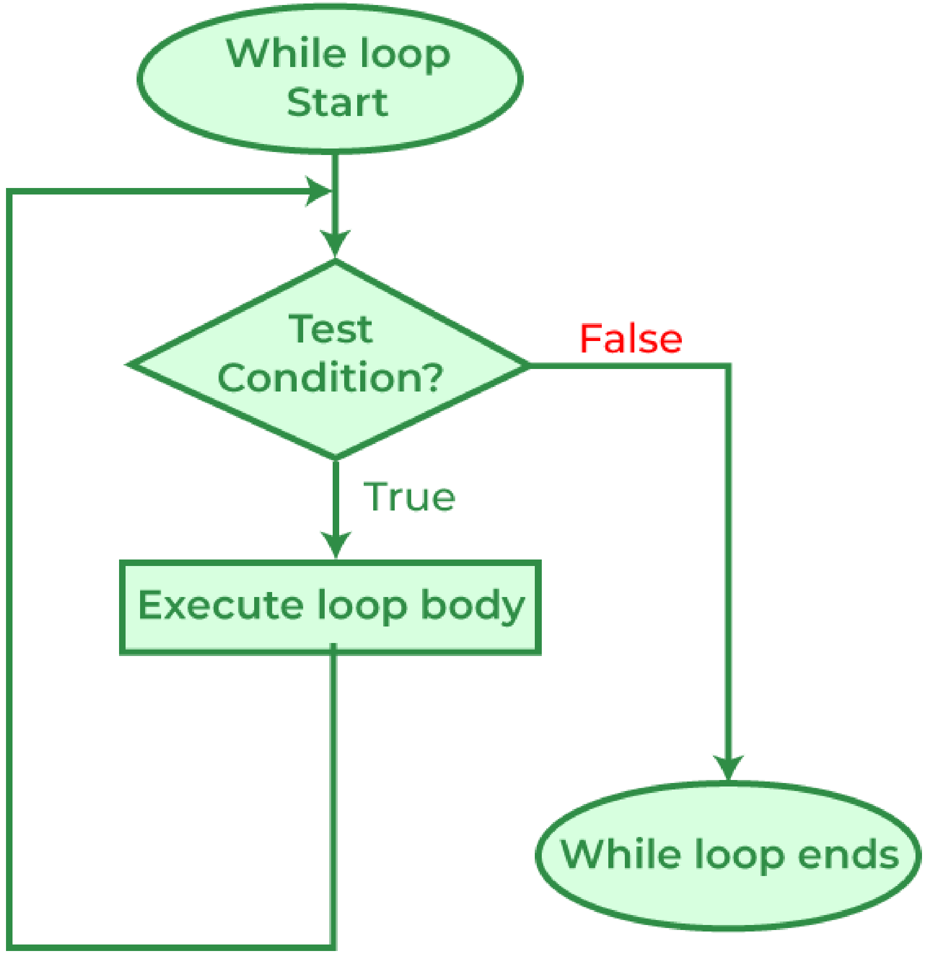
- Control falls into the while loop.
- The flow jumps to Condition
- Condition is tested.
  - If the Condition yields true, the flow goes into the Body.
  - If the Condition yields false, the flow goes outside the loop
- The statements inside the body of the loop get executed.
- Updation takes place.
- Control flows back to Step 2.
- The while loop has ended and the flow has gone outside.

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

int main() {
		int i = 0;
		while(i<11) {
				cout<<i<<end;
				i++;
		}
}
```

The while Loop

- **Flow Chart of while loop**
  1. Start: Begin the process or program.
  2. Initialize Variables: Set up any variables required for the loop.
  3. Condition Check:
     - Evaluate the condition inside the while loop.
     - If true, proceed to the next step.
     - If false, exit the loop.
  4. Execute Code Block: Run the code inside the loop.
  5. Update Variables:
     - Update the loop control variable (e.g., increment or modify it).
     - This step ensures the condition will eventually become false to avoid an infinite loop.
  6. Repeat: Go back to Step 3 (Condition Check) and repeat the process.
  7. Exit Loop: When the condition becomes false, exit the loop and continue with the rest of the program.



The while Loop

Code Example

Print the multiplication table for 5.

1_Basic_Program.cpp
2_sum_of_n_natural_num.cpp
3_Validate_User_Input.cpp
4_Multiplication_Table_of_a_Number.cpp
5_Factoiral_of_Number.cpp
6_SImple_ATM_Withdrawal.cpp
7_Some_Tricky_Program.cpp
8_Power_of_Number.cpp

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

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

Thank You