

Lecture 08

M M Imran

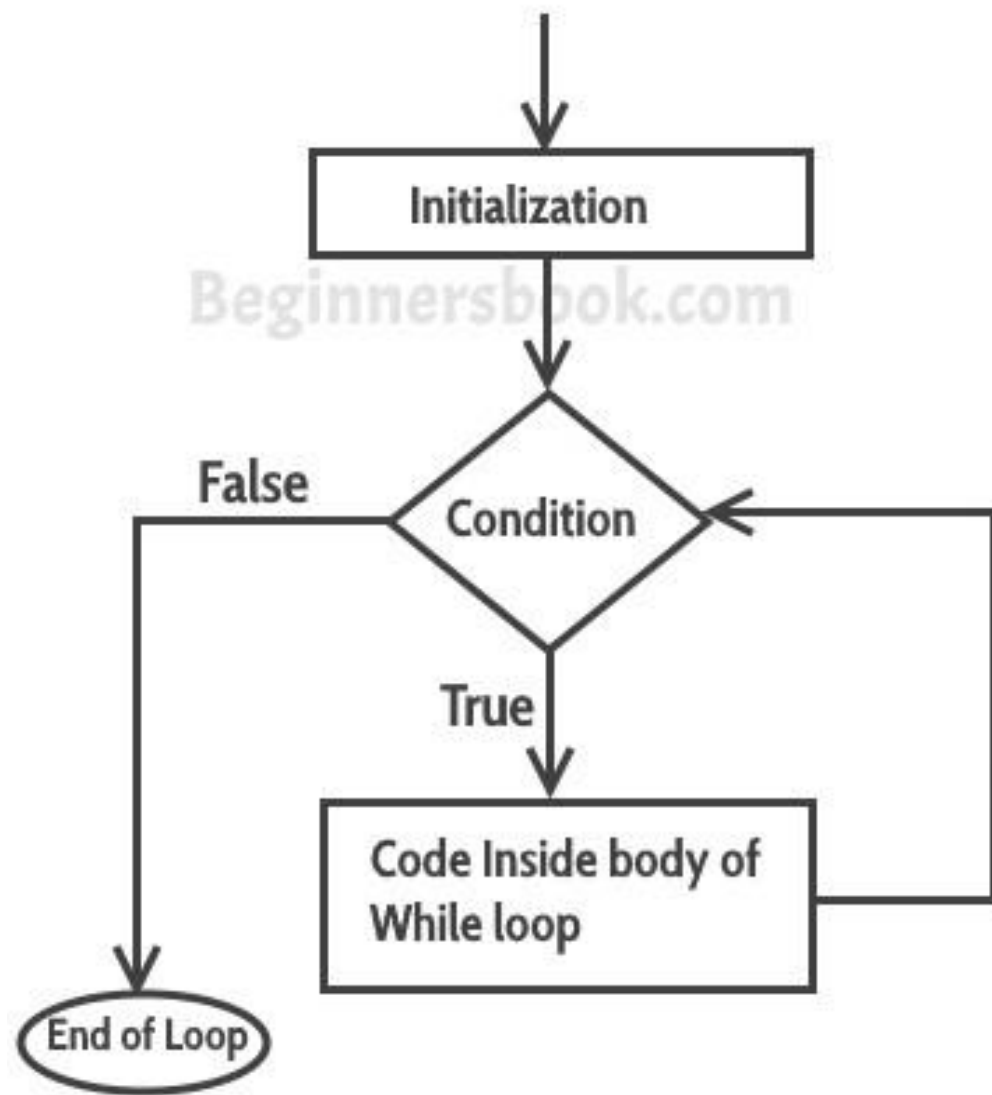
while Loop

- As discussed earlier, loops are used for executing a block of program statements repeatedly until the given loop condition returns false.
- The while statement is an iterative statement with a condition and a block of code.
- A while statement is a pre-test loop: the condition is tested before the block of code executes.
- A while statement does indefinite iteration: the number of times it loops is unknown ahead of time.
- A while statement loops zero or more times.
- A while statement continues to execute the block of code while the condition is true.

while Loop Syntax

```
while (<condition>) {  
    <block>  
}
```

- A **while** loop evaluates the condition
- If the **condition** evaluates to **true**, the code inside the **while** loop is executed.
- The **condition** is evaluated again.
- This process continues until the **condition** is **false**.
- When the **condition** evaluates to **false**, the loop terminates.



```
#include <iostream>

using namespace std;

int main() {
    int i = 1;

    // while loop from 1 to 5
    while (i <= 5) {
        cout << i << " ";
        i++; // increment loop invariant
    }

    return 0;
}
```

Output

```
1 2 3 4 5
```

Explanation

Iteration	Variable	$i \leq 5$	Action
1st	$i = 1$	true	1 is printed and i is increased to 2.
2nd	$i = 2$	true	2 is printed and i is increased to 3.
3rd	$i = 3$	true	3 is printed and i is increased to 4
4th	$i = 4$	true	4 is printed and i is increased to 5.
5th	$i = 5$	true	5 is printed and i is increased to 6.
6th	$i = 6$	false	The loop is terminated

Printing Even Numbers

```
int main() {  
    int i = 0;  
    int number;  
  
    // take input from the user  
    cout << "Enter a number: ";  
    cin >> number;  
  
    while (i <= number) {  
        cout << i << " ";  
        i = i + 2;  
    }  
  
    return 0;  
}
```

```
Enter a number: 10  
0 2 4 6 8 10
```

Printing Even Numbers (2)

```
int main() {  
    int i = 0;  
    int number;  
  
    // take input from the user  
    cout << "Enter a number: ";  
    cin >> number;  
  
    while (i <= number) {  
        if (i % 2 == 0) {  
            cout << i << " ";  
        }  
  
        i = i + 1;  
    }  
  
    return 0;  
}
```

```
Enter a number: 10  
0 2 4 6 8 10
```


Infinite while Loop

- If the condition of a loop is always true, the loop runs for infinite times (until the memory is full).
- It could be a case that programmer forget to increment / decrement correctly.
- Or it could be a case that the condition written never get false.

What's the output of this code?

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

int main() {

    int i = 1;

    while (i < 5) {
        cout << i << " ";
    }

    return 0;
}
```

What about this one?

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

int main() {

    int i = 15;

    while (i > 5) {
        cout << i << " ";
        i++;
    }

    return 0;
}
```

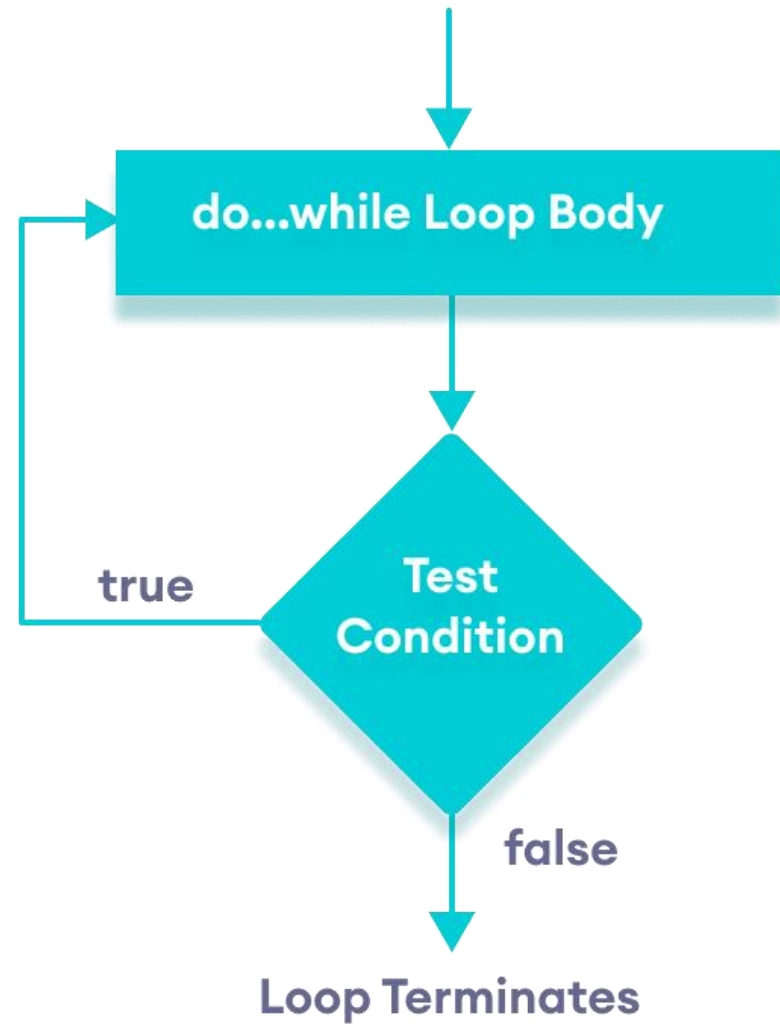
do-while Loop

- A do-while statement is an iterative (looping) statement with a condition and a block of code.
- A do-while statement is a post-test loop: the condition is tested after the block of code executes.
- A do-while statement does indefinite iteration: the number of times it loops is unknown ahead of time.
- A do-while statement loops one or more times.
- A do-while statement continues to execute the block of code while the condition is true.

do-while Loop Syntax

```
do {  
    <block>  
} while (<condition>)
```

- The body of the loop is executed at first. Then the condition is evaluated.
- If the condition evaluates to true, the body of the loop inside the do statement is executed again.
- The condition is evaluated once again.
- If the condition evaluates to true, the body of the loop inside the do statement is executed again.
- This process continues until the condition evaluates to false. Then the loop stops.



```
#include <iostream>

using namespace std;

int main() {
    int i = 1;

    // do...while loop from 1 to 5
    do {
        cout << i << " ";
        ++i;
    }
    while (i <= 5);

    return 0;
}
```

Output

```
1 2 3 4 5
```

```
#include <iostream>

using namespace std;

int main() {
    int i = 15;

    do {
        cout << i << " ";
        ++i;
    } while (i <= 5);

    return 0;
}
```

Output:
15

Infinite do-while Loop

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

int main() {
    do {
        cout<<"Infinitive do-while Loop";
    } while(true);

    return 0;
}
```

for vs while loops

- A for loop is usually used when the number of iterations is known.
- However, while and do...while loops are usually used when the number of iterations is unknown.
- In other words, you should use a for loop when you know how many times the loop should run. If you want the loop to break based on a condition other than the number of times it runs, you should use a while loop.

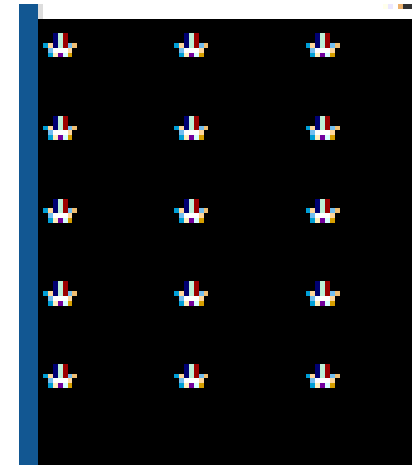
Displaying a Pattern (while Loop)

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

int main() {
    int i = 1, j = 1;
    int rows = 5;
    int columns = 3;

    while (i <= rows) {
        j = 1;
        while (j <= columns) {
            cout << "* ";
            j++;
        }
        cout << endl;
        i++;
    }

    return 0;
}
```



Displaying a Pattern (for Loop)

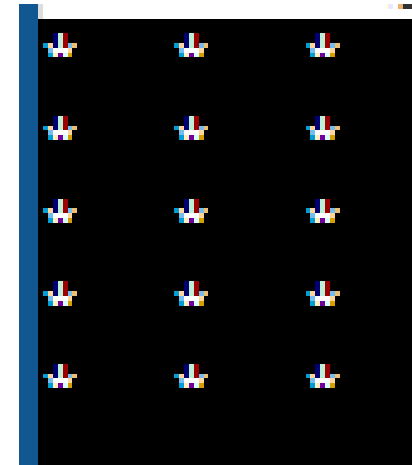
```
#include <iostream>
using namespace std;

int main() {

    int rows = 5;
    int columns = 3;

    for (int i = 1; i <= rows; i++) {
        for (int j = 1; j <= columns; j++) {
            cout << "* ";
        }
        cout << endl;
    }

    return 0;
}
```



Find the first and last digit of a number

Example

Input

Input number: 1234

Output

First digit: 1

Last digit: 4

Logic

Finding last digit:

We use modulo operator %. When modulo divided by 10 returns its last digit.

Suppose if $n = 1234$ then last Digit = $n \% 10 \Rightarrow 4$

Finding first digit:

To find first digit of a number we divide the given number by 10 until number is greater than 10

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cout << "Input number: ";
    cin >> n;
    int temp = n;

    while (temp > 10) {
        temp = temp / 10;
    }

    cout << "First digit: " << temp << endl;
    cout << "Last digit: " << n % 10 << endl;

    return 0;
}
```

```
Input number: 123
First digit: 1
Last digit: 3
```

Thanks !!

