Week 6

Session 6: Repetitive tasks using 'for' loop statement

Element 4: Plan and execute C++ applications using loop structures, 'for' and 'while' statements

ECT 124: Writing Programs using C++

Performance criteria (PC) for E4

PC1: Write code for repetitive tasks using 'for' loops.

In this lesson!

PC2: Write code for repetitive tasks using 'while' and 'Do-while' statements.

Learning objectives:

By the end of this lesson, the student should be able to:

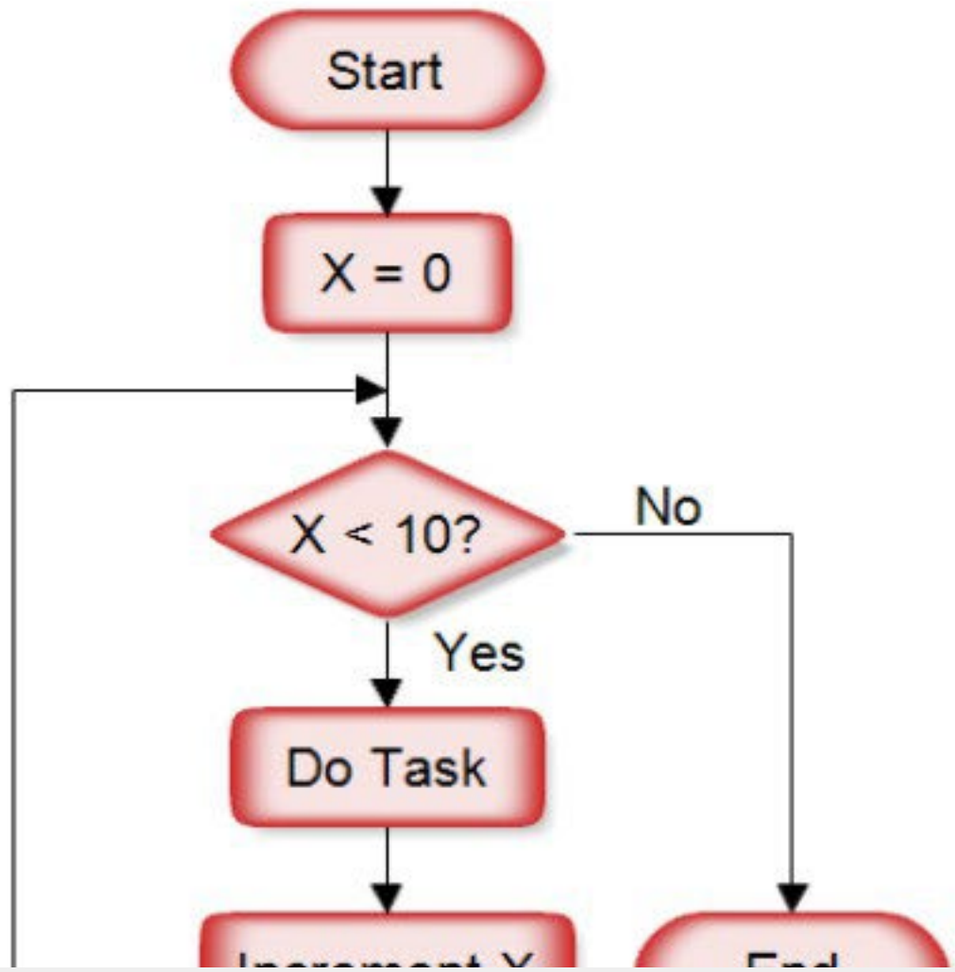
- ✓ Write applications in C++ using 'for' loop statement.

Class Activity 1

Draw It

Using a blue pen, circle the flowchart section that indicates a loop statement procedure.

Instructions



Submit

Loop Statements

- Although we can solve problems using the sequence and selection/branching statements as shown in previous lessons, this approach will not solve all problems effectively.

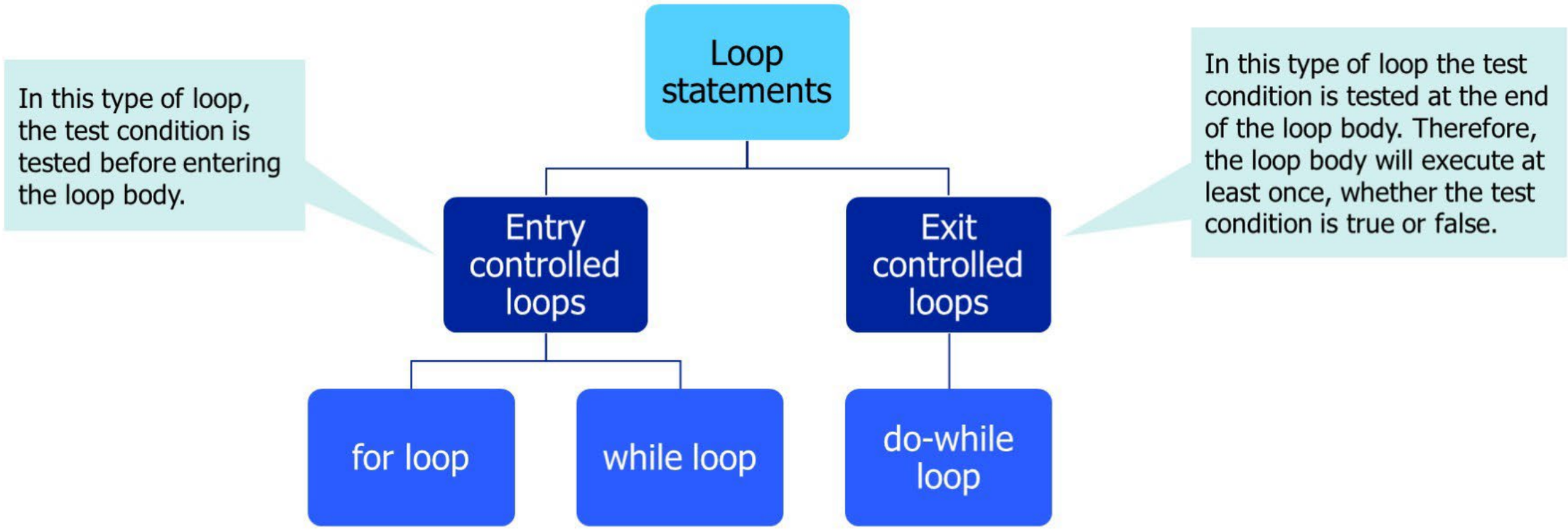
- Consider we want to change 50 test scores to their associating letter grades. We have 3 choices:

1) We can write a program with only one selection statement and run it 50 times. This approach is very time consuming.
2) We can write a program with 50 selection statements. This approach is long and must be changed if the number of students changes.
3) We can use loop statement that allow us to repeat one activity or a set of activities as many times as we desire.

- When solving a problem case similar to above, we have the option to repeat lines of code.

- Repetition allows the C++ program to iterate a section of code multiple times. **This repetition of codes is called loop statement**. A loop is a sequence of instructions that is repeated until a certain condition is reached

- Loops come into use when we need to repeatedly execute a block of statements. For the loop statements, there are two types of loop as depicted in the figure below.



- The for loop and while loop are example of entry controlled loops whereas do-while loop is categorized as exit controlled loops.

Loop type		Description
Entry controlled loop	for loop	Firstly initializes, then, condition check, execute body, update.
	while loop	First checks the condition, then executes the body.
Exit controlled loop	do-while loop	Firstly, execute the body then condition check.

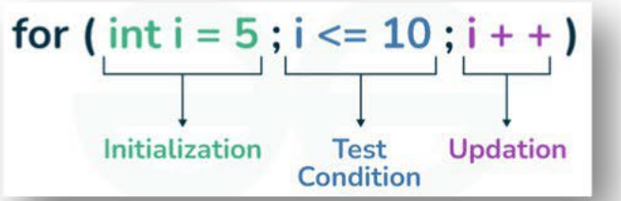
- In any loop statements, we utilize a counter to determine how many times the body of the loop must be repeated.
- We can set the counter to an initial value before the loop, increment or decrement the counter in each iteration, and stop the loop when we are done.

The for Loop Statements

- In C++, **for loop** is an entry-controlled loop that is used to execute a block of code repeatedly for the specified range of values. Basically, it allows you to repeat a set of instructions for a specific number of iterations.

- The for loop is generally preferred over while and do-while loops in case the number of iterations is known beforehand. The for loop syntax is:

```
for (initialization; condition; update)
{
// statement -1;
// statement -n;
}
```



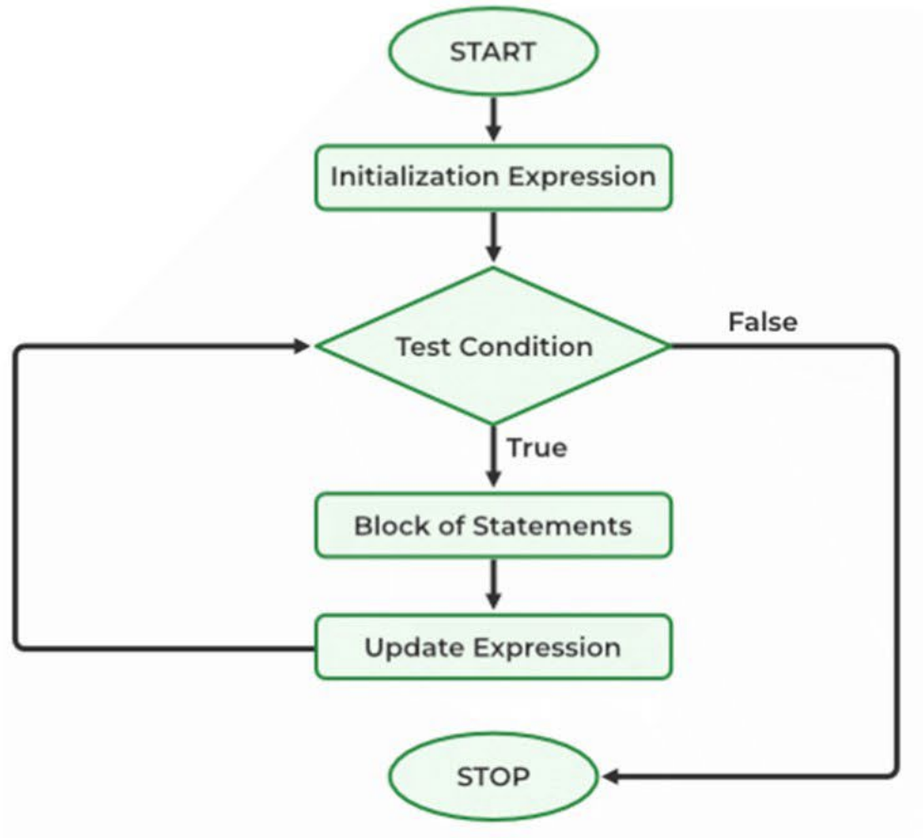
initialization	Initializes variables and is executed only once
condition	If true, the body of for loop is executed or if false, the for loop is terminated
update	Updates the value of initialized variables and again checks the condition

- The for loop statement is useful when we need counter-controlled iteration. It combines three elements of a loop — initialization, conditional test, and update — into the loop construct itself.

- The for loop repeats statement (or called loop body) while condition is true.

- It provides specific initialization and an increase expression, executed before the loop begins and after each iteration, respectively. Therefore, it is useful to use counter variables as condition.

- The flowchart and execution flow of for loop are described below:



Execution Flow of a 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. Updation takes place and the flow goes to Step 3 again.
7. The for loop has ended and the flow has gone outside.

Example 1 Printing Numbers from 1 to 5

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

int main()
{
    for (int i = 1; i <= 5; i = i + 1)
    {
    cout << "Value of i: "<< i << endl;
    }

    return 0;
}
```

```
C:\Users\msharizal\OneDrive - Higher Co
Value of i: 1
Value of i: 2
Value of i: 3
Value of i: 4
Value of i: 5

--------------------------------
Process exited after 0.05158 se
Press any key to continue . . .
```

For the counter, we can also utilize postfix increment (i++) rather than then conventional adding of i = i +1

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

int main()
{
    for (int i = 1; i <= 5; i++)
    {
    cout << "Value of i: "<< i << endl;
    }

    return 0;
}
```

```
C:\Users\msharizal\OneDrive - Higher C
Value of i: 1
Value of i: 2
Value of i: 3
Value of i: 4
Value of i: 5

--------------------------------
Process exited after 0.06696 s
Press any key to continue . .
```

Example 2 Printing Numbers from 10 to 15

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

int main()
{
    int n = 15, a;

    for (a = 10; a <= n; a++)
    {
        cout << a << " ";
    }

    return 0;
}
```

```
C:\Users\msharizal\OneDrive - Higher C
10 11 12 13 14 15
--------------------------------
Process exited after 0.3401 se
Press any key to continue . .
```

The above program uses a for loop to print numbers from 10 to n (here n=15). The loop variable a iterates from 10 to n and in each iteration condition is checked (is a<=n i.e., i<=15), if true then it prints the value of a followed by a space and increment a. When the condition is false loop terminates.

Example 3 Display a text for 10 times

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

int main()
{
    for (int i = 1; i <= 10; i++)
    {
    cout << "Hello ECT 124" << endl;
    }

    return 0;
}
```

```
C:\Users\msharizal\OneDrive - H
Hello ECT 124
Hello ECT 124
Hello ECT 124
Hello ECT 124
Hello ECT 124
Hello ECT 124
Hello ECT 124
Hello ECT 124
Hello ECT 124
Hello ECT 124

--------------------------
Process exited after 0.10
Press any key to continue
```

Example 4 Find the sum of first 7 natural numbers

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

int main()
{
    int i, sum = 0;

    for (i = 1; i <= 7; i++)
    {
    cout << i << " ";
    sum = sum + i;
    }
    cout << "\nThe sum of first 7 natural numbers: " << sum << endl;

    return 0;
}
```

```
C:\Users\msharizal\OneDrive - Higher Colleges of Techr
1 2 3 4 5 6 7
The sum of first 7 natural numbers: 28

--------------------------------
Process exited after 0.03508 seconds wit
Press any key to continue . . .
```

$1 + 2 + 3 + 4 + 5 + 6 + 7 = 28$

Example 5 Find the number and sum of all integer between 1 and 20 which are divisible by 2.

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

int main()
{
    int i, sum = 0;

    cout << "Numbers between 1 and 20, divisible by 2: " << endl;
    cout << "-------------------------------------------: " << endl;

    for (i = 2; i < 20; i++)
    {
        if (i % 2 == 0)
        {
            cout << " " << i;
            sum += i;
        }
    }

    cout << "\n The sum : " << sum << endl;

    return 0;
}
```

```
C:\Users\msharizal\OneDrive - Higher Colleges of Technology\M
Numbers between 1 and 20, divisible by 2:
-------------------------------------------
 2 4 6 8 10 12 14 16 18
 The sum : 90

--------------------------------
Process exited after 0.2428 seconds with retur
Press any key to continue . . .
```

$2 + 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18 = 90$

Nested for Loop Statements

- A **nested for loop** is basically putting one for loop inside another for loop. Every time the outer loop runs, the inner loop runs altogether. It is a way to repeat tasks within tasks in the program.

Example 6 Nested loop demonstration.

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

int main()
{
    int i, j;

    // Outer loop
    for (int i = 0; i < 4; i++)
    {
        // Inner loop
        for (int j = 0; j < 4; j++)
        {
            // Printing the value of i and j
            cout << "*" << " ";
        }
        cout << endl;
    }
}
```

```
C:\Users\msharizal\OneDrive - Higher C
* * * *
* * * *
* * * *
* * * *

--------------------------------
Process exited after 0.01694 s
Press any key to continue . .
```

The above program uses nested for loops to print a 4×4 matrix of asterisks (*). Here, the outer loop (i) iterates over rows and the inner loop (j) iterates over columns. In each iteration, inner loop prints an asterisk and a space also new line is added after each row is printed.

Class Activity 2

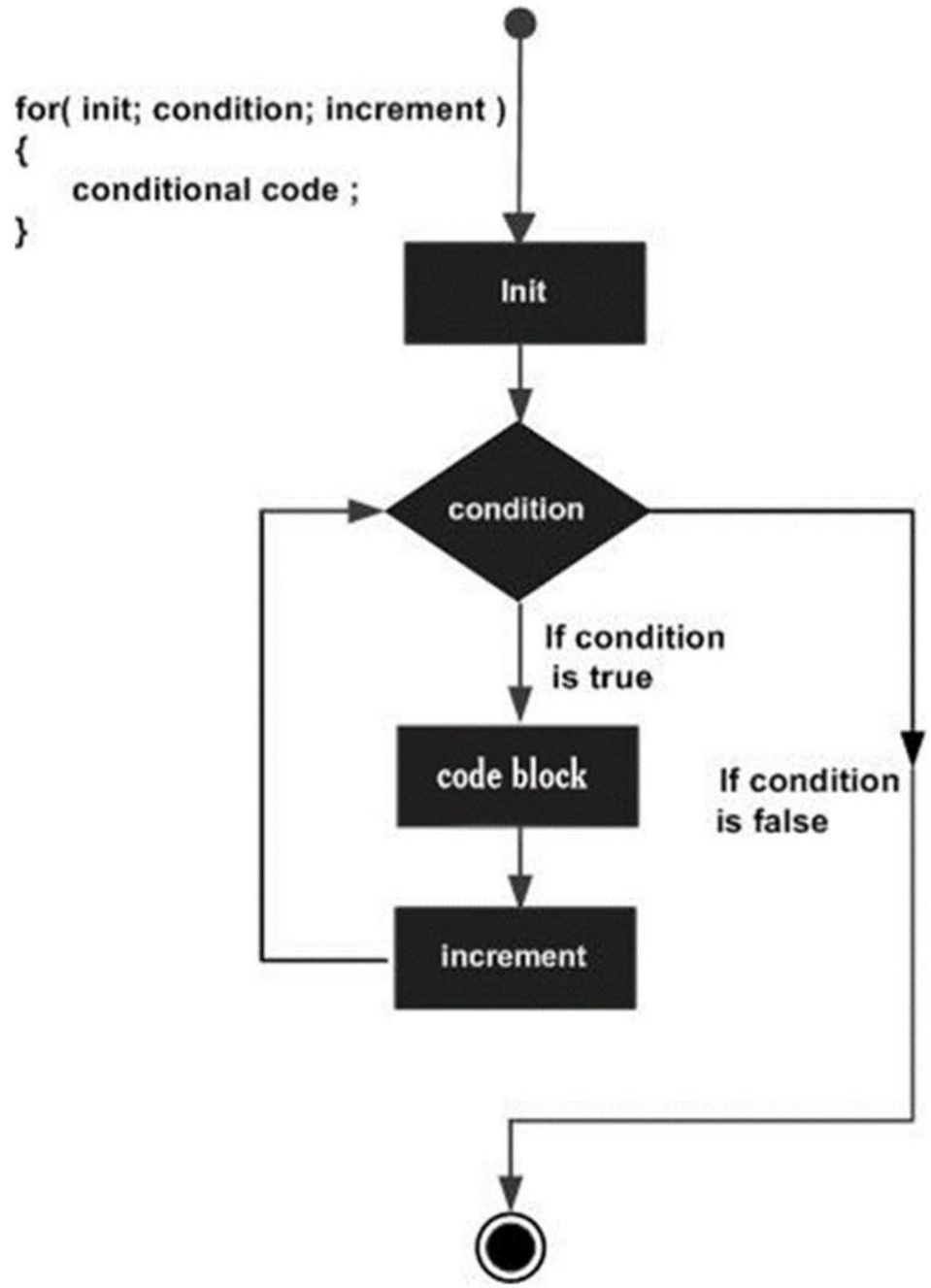
Loops in C++(DO WHILE AND FOR -17)

Quiz

The for loop statement is an exit controlled type of loop.

- TRUE
- FALSE

Summary


for(init; condition; increment)
{
conditional code ;
}
Init
condition
If condition
is true
code block
If condition
is false
increment

Higher Colleges of Technology

كليات التقنية العليا

Thank You

800 MyHCT (800 69428) www.hct.ac.ae HCT_UAE | hctuae