LAB #05: INTRODUCTION TO FOR LOOPS, SWITCH CASE

Lab Objective:

To practice for loops and switch cases and to get better understanding of how to use them.

Lab Description:

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

Syntax:

The syntax of a for loop in C++ is −

```
for ( init; condition; increment )
{
    statement(s);
}
```
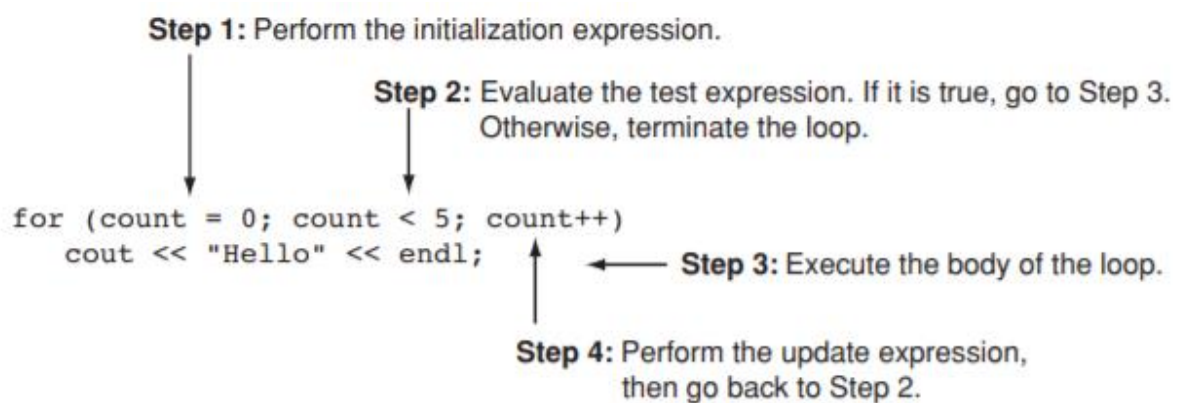
Step 1: Perform the initialization expression.

Step 2: Evaluate the test expression. If it is true, go to Step 3. Otherwise, terminate the loop.

```
for (count = 0; count < 5; count++)
    cout << "Hello" << endl;
```

Step 3: Execute the body of the loop.

Step 4: Perform the update expression, then go back to Step 2.

Figure 2: the loop's logic.

Example

```
int main (){

const int MIN_NUMBER = 1,    // Starting value
          MAX_NUMBER = 10;   // Ending value
int num;

cout << "Number Number Squared\n";
cout << "-------------------------\n";

for (num = MIN_NUMBER; num <= MAX_NUMBER; num++)
    cout << num << "\t\t" << (num * num) << endl;

return 0;

}
```

Output:

```
Number  Number  Squared
----------------------
1                  1
2                  4
3                  9
4                  16
5                  25
6                  36
7                  49
8                  64
9                  81
10                 100
```

Other Forms of the Update Expression :

You are not limited to using increment statements in the update expression. Here is a loop that displays all the even numbers from 2 through 100 by adding 2 to its counter:

```
for (num = 2; num <= 100; num += 2)
    cout << num << endl;
```

And here is a loop that counts backward from 10 down to 0:

```
for (num = 10; num >= 0; num--)
    cout << num << endl;
```

Defining a Variable in the for Loop's Initialization Expression:

Not only may the counter variable be initialized in the initialization expression, it may be defined there as well. The following code shows an example.

```
for (int num = MIN_NUMBER; num <= MAX_NUMBER; num++)
    cout << num << "\t\t" << (num * num) << endl;
```

Using Multiple Statements in the Initialization and Update Expressions:

It is possible to execute more than one statement in the initialization expression and the update expression. When using multiple statements in either of these expressions, simply separate the statements with commas. For example

```
int x, y;
for (x = 1, y = 1; x <= 5; x++)
{
    cout << x << " plus " << y
         << " equals " << (x + y)
         << endl;
}
```

Omitting the for Loop's Expressions:

The initialization expression may be omitted from inside the for loop's parentheses if it has already been performed or no initialization is needed. Here is an example

```
int num = 1;
for ( ; num <= maxValue; num++)
    cout << num << "\t\t" << (num * num) << endl;
```

Nested Loops:

A loop that is inside another loop is called a nested loop. A clock is a good example of something that works like a nested loop. The second hand, minute hand, and hour hand all spin around the face of the clock. Each time the hour hand increments, the minute hand increments 60 times. Each time the minute hand increments, the second hand increments 60 times. Here is a program segment with a for loop that partially simulates a digital clock. It displays the seconds from 0 to 59:

Example

```
for (int hours = 0; hours < 24; hours++)
{
    for (int minutes = 0; minutes < 60; minutes++)
    {
        for (int seconds = 0; seconds < 60; seconds++)
        {
            cout << setw(2) << hours << ":";
            cout << setw(2) << minutes << ":";
            cout << setw(2) << seconds << endl;
        }
    }
}
```

Output:

```
00:00:00
00:00:01
00:00:02
    .       (The program will count through each second of 24 hours.)
    .
    .
23:59:59
```

Switch case

switch...case is a branching statement used to perform action based on available choices, instead of making decisions based on conditions. Using switch...case you can write more clean and optimal code than if...else statement. switch...case only works with integer, character and enumeration constants.

Lab Tasks:

- Task1: C++ program to Print Table of any Number.

The concept of generating **table of any number** is multiply particular number from 1 to 10.

num	*	1
num	*	2
num	*	3
num	*	4
num	*	5
num	*	6
num	*	7
num	*	8
num	*	9

num * 10

- Task2: Find power of any number using for loop.

- Task3: Sum of Natural Numbers using loop.

- Task4:C++ program to find sum of digits of a number.

    Sum of digits means add all the digits of any number, for example we take any number like 358. Its sum of all digit is 3+5+8=16. Using given code we can easily write c++ program.

- Task 5: Program to Generate Factorial a Certain Number

    Factorial of any number is the product of an integer and all the integers below it for example factorial of 4 is 4! = 4 * 3 * 2 * 1 = 24

- Task7: Write a C++ program that will print the following pattern.
    ```
    *
    **
    ***
    ****
    *****
    ******
    *******
    ```
- Write a C++ program print total number of days in a month using switch case.
- Write a C++ program to check whether an alphabet is vowel or consonant using switch case.

Think??

- What will be the output of the C++ program?

```cpp
#include<iostream>
using namespace std;
int main()

{

    for(5;2;2)

    {

            cout<<"Hello";

    }

    return 0;

}
```

- What will be the output of the C++ program, if input is 6?

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

int main()
{
        int i;

        for(i = 0; i>9; i+=3)
        {
                cout<<"for ";
        }
        return 0;

}
```

- What will be the output of the C ++ program.

```cpp
#include<iostream>
using namespace std;
int main()
{
        for(;;)
        {
                cout<<"10";
        }
        return 0;

}
```

- What will be the output of the C++ program?

```cpp
int main()
{
        int fun=5;
        cout<<"C++ for loop ";
        int x = 5;
        for(x=0;x<=fun;x++)
```

```
        {
                cout<<x;
        }
        return 0;
}
```