Fundamental of Programing

Lab Manual # 04

**Course Instructor:** Dr Jawad Khan

**Lab Instructor:** Muhammad Affan

## Student Name:

## CMS ID:

**DATE:**

**Lab Manual # 04**

**Loops (Part-01)**

# Objective:

To understand repetition structure and the types of repetition structure.

# Description:

Repetition structures, or loops, are used when a program needs to repeatedly process one or more instructions until some condition is met, at which time the loop ends.

## Types of Repetition Structures:

Two types of repetition structures: 1) Pretest and 2) post-test loops.

### Pretest:

Loop condition appears at the beginning of the pretest loop.

Determines the number of times instructions w/in loop body are processed.

Types of pretest loops are:

* while
* for

**Post-test:**

Loop condition appears at end of posttest loop.

Determines the number of times instructions w/in loop body are processed 3. HOWEVER, instructions are processed at least once--the first time!

Types of posttest loop are:

* do...while Loop

# For Loop

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 −

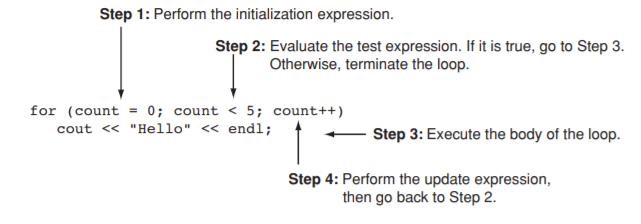
for ( init; condition; increment )

{

statement(s);

}

## How it works



## Explanation of the Syntax:

* **Initialization statement:**

This statement gets executed only once, at the beginning of the for loop. You can enter a declaration of multiple variables of one type, such as int x=0, a=1, b=2. These variables are only valid in the scope of the loop. Variable defined before the loop with the same name are hidden during execution 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.)

## NOTES:

* The initialization and increment statements can perform operations unrelated to the condition statement, or nothing at all – if you wish to do. But the good practice is to only perform operations directly relevant to the loop.
* A variable declared in the initialization statement is visible only inside the scope of the for loop and will be released out of the loop.
* Don’t forget that the variable which was declared in the initialization statement can be  modified during the loop, as well as the variable checked in the condition.

## Flow Diagram of Loop:

for loop in C++

## Example:

|  |  |
| --- | --- |
| **Example** |  |
| int main (){  } | **Output:** |

### 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:



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



# Lab Task:

1. Write a program in C++ to find the sum of first 10 natural numbers.
2. Write a C++ program to Print a Table of any Number.

The output should look like this:

1. Write a 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
2. Write a C++ program to generate a Fibonacci sequence up to a certain number input by the user.
3. Write a C++ program to print the full pyramid using \* where the height of the pyramid is input by the user using for loop.

Example of a full pyramid:



# Home Task:

1. Write a program in C++ that prints the numbers from 1 to 150 except the multiples of 10.

Make use of the continue statement.

1. Write a C++ program to find the sum of digits of a number.

The sum of digits means adding all the digits of any number, for example, we take any number like 358. Its sum of all digits is 3+5+8=16.

1. Write a program in C++ to check whether a number is prime or not.