National University of Computer and Emerging Sciences



Lab Manual

for

Programming Fundamentals

|  |  |
| --- | --- |
| Course Instructor | Mr. Owais Idrees |
| Lab Instructor(s) | Muntaha Zaigham  Abiha Aftab |
| Section | PF J |
| Semester | Fall 2021 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

# INTRODUCTION

**DO-WHILE LOOP:**

A **do-while** loop statement repeatedly executes a target statement as long as a given condition is true. In this loop, the condition comes after the body of the loop. The loop is important in a situation where a statement must be executed at least once.

## Syntax

The syntax of a do-while loop in C++ is −

do{

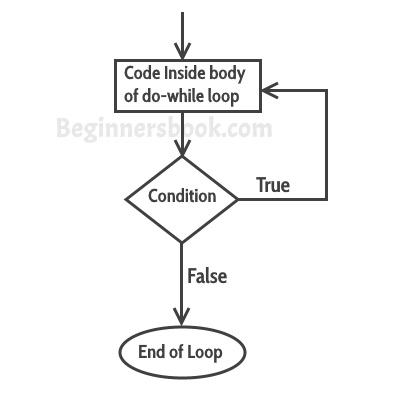
statement(s);

} while(condition);

Here, **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any non-zero value. The loop iterates while the condition is true.

When the condition becomes false, program control passes to the line immediately following the loop.

## Flow Diagram



Here, key point of the do-*while* loop is that the loop body runs once and then the condition is checked. When the condition is tested and the result is false, the first statement after the while loop will be executed else if the result is true the loop body will get executed.

Example:

Output:

Hello

Hello

Hello

Hello

Hello

Hello

int n;

n =1;

do

{

cout<<”Hello”<<endl;

n++;

}

while(n<=5);

# Lab Manual 06

**Problem 1:**

Write a program that reads the current state of telephone line. The user should enter w for working and d for dead state. Any other input should be considered invalid. Use do-while loop to force the user to enter a valid input value.

**Sample Output:**

Press w for working state and d for dead state

Enter the state of your phone: g

Enter the state of your phone: 2

Enter the state of your phone: d

Noted, your phone is in dead state.

## Problem 2 : Write an infinite loop that never ends. Condition is always true.

## Problem 3:

Write a program to find the sum of odd and even numbers from 1 to N.

**Sample Output:**

Enter the value of num 10

Sum of all odd numbers = 25

Sum of all even numbers = 30

**Problem 4:**

Two numbers are **amicable** if the first is equal to the sum of divisors of the second, and if the second number is equal to the sum of divisors of the first. Write a program that finds whether two numbers entered by user are amicable or not.

Hint:

 One can easily check that the sum of the proper divisors of 284 is 1 + 2 + 4 + 71 + 142 = 220 and the sum of the proper divisors of 220 is 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284. So 284 and 220 are amicable.

**Problem 5:**

Let’s design a game of Rock-Paper-Scissors in C++. There are 2 players and both of them simultaneously say their words ('P' for paper, 'S' for scissors, and ‘R’ for rock). The rules of the game are fairly simple.

* Rock dominates Scissors
* Paper dominates Rock
* Scissors dominate Paper

If player-1 says **‘P**’ and player-2 says **‘R’**, you'll print "**Player 1 wins**". Likewise if player-1 says **‘P’** and player-2 says **‘S**’, you'll print "**Player 2 wins**" and so on. If both the players say the same word, it's a **draw**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Player-2** | | | |
| **Player-1** |  | ***Rock*** | ***Paper*** | ***Scissors*** |
| ***Rock*** | Draw | Player-2 Wins | Player-1 Wins |
| ***Paper*** | Player-1 Wins | Draw | Player-2 Wins |
| ***Scissors*** | Player-2 Wins | Player-1 Wins | Draw |