

LAB #06: While & Do While Loops

Name:_____

Reg #:_____

Lab Objective:

To have better understanding regarding loops.

Lab Description:

A loop is part of a program that repeats. The while loop has two important parts: (1) an expression that is tested for a true or false value, and (2) a statement or block that is repeated as long as the expression is true.

```
while (expression)
{
    statement;
    statement;
    // Place as many statements here
    // as necessary.
}
```

The while Loop Is a Pretest Loop ,which means it tests its expression before each iteration whereas the do-while loop is a posttest loop, which means its expression is tested after each iteration.

```
do
{
    statement;
    statement;
    // Place as many statements here
    // as necessary.
} while (expression);
```

Infinite Loops:

If a loop does not have a way of stopping, it is called an infinite loop. An infinite loop continues to repeat until the program is interrupted. Here is an example of an infinite loop:

```
int number = 0;
while (number < 5)
{
    cout << "Hello\n";
}
```

We can make this loop finite by adding a line as shown below

```
while (number < 5)
{
    cout << "Hello\n";
    number++;
}
```

Examples: The following example averages a series of three test scores for a student. After the average is displayed, it asks the user if he or she wants to average another set of test scores. The program repeats as long as the user enters Y for yes.

Example	OUTPUT
<pre>#include <iostream> using namespace std; int main() { int score1, score2, score3; // Three scores double average; // Average score char again; // To hold Y or N input do { // Get three scores. cout << "Enter 3 scores and I will average them: "; cin >> score1 >> score2 >> score3; // Calculate and display the average. average = (score1 + score2 + score3) / 3.0; cout << "The average is " << average << ".\n"; // Does the user want to average another set? cout << "Do you want to average another set? (Y/N) "; cin >> again; } while (again == 'Y' again == 'y'); return 0; }</pre>	<pre>Enter 3 scores and I will average them: 80 90 70 [Enter] The average is 80. Do you want to average another set? (Y/N) y [Enter] Enter 3 scores and I will average them: 60 75 88 [Enter] The average is 74.3333. Do you want to average another set? (Y/N) n [Enter]</pre>

Lab Tasks:

- Task1: program to print all natural numbers from 1 to n using while loop
- Task2: program to print natural numbers in reverse from n to 1 using while loop
- Task3: program to print all even numbers between i to n using while loop.
- Task4: Program to Generate Fibonacci Sequence up to a Certain Number. The Fibonacci sequence is a series of numbers where a number is found by adding up the two numbers before it. Starting with 0 and 1, the sequence goes 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, and so forth.
- Task 5: Take i and n two inputs from user and print n^i of the number.
- Task6: Write a method with a while loop that computes the sum of first n positive integers: $\text{sum} = 1 + 2 + 3 + \dots + n$.
- Task7: Write a do-while loop that asks the user to enter two numbers. The numbers should be added and the sum displayed. The user should be asked if he or she wishes to perform the operation again. If so, the loop should repeat; otherwise it should terminate.
- Task8: Convert the following while loop to a do-while loop:

```
int x = 1;
while (x > 0)
{
    cout << "enter a number: ";
    cin >> x;
}
```

Think??

- What will the following program segments display?

```
A) int count = 10;
    do
    {
        cout << "Hello World\n";
        count++;
    } while (count < 1);
```

```

B) int v = 10;
    do
        cout << v << endl;
    while (v < 5);

C) int count = 0, number = 0, limit = 4;
    do
    {
        number += 2;
        count++;
    } while (count < limit);
    cout << number << " " << count << endl;

```

- What's wrong with the following while loop?

```

int counter = 0;
while {counter > 100}
    if (counter % 2 == 1)
        cout << counter << " is odd." << endl;
    else
        cout << counter << " is odd." << endl;
    ++counter;    // same as: counter = counter + 1;

```