| ***Computer Engineering Department*** |
| --- |
| ***CE100L: Computing Fundamentals & Programming*** |

| ***Course Instructor: Usama Bin Shakeel*** | ***Dated: 25/10/2021*** |
| --- | --- |
| ***Teaching Assistant: Aqsa Khalid*** | ***Semester: Fall 2021*** |
| ***Lab Engineer: Nadir Abbas*** | ***Batch: BSCE2021*** |

# **Lab 5A. Implementation of Do While Loop**

| **Name** | **Roll number** | **Report**  **(out of 100)** | **Scaled to 10** | **Total**  **(out of 10)** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Checked on: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Objective**

The goal of this handout is to learn the working of loops in C++.

## **Equipment and Component**

| **Component Description** | **Value** | **Quantity** |
| --- | --- | --- |
| Computer | Available in lab | 1 |

## **Conduct of Lab**

1. Students are required to perform this experiment individually.
2. In case the lab experiment is not understood, the students are advised to seek help from the course instructor, lab engineers, assigned teaching assistants (TA) and lab attendants.

## **Theory and Background**

A statement or a set of statements that is executed repeatedly is called a loop. The statement(s) in a loop are executed for a specified number of times or until some given condition remains true. In C++, there are three kinds of loop statements. These are: The “for” loop. The “while” loop. The “do-while” loop.

The “while” loop:- It is a conditional loop statement. It is used to execute a statement or a set of statements as long as the given condition remains true.

In for loop the number of iterations to be done is already known. In other words, it is used to execute a statement or a set of statements for a fixed number of times.

The “do-while” loop is also a conditional loop statement. It is like a while loop but in this loop the condition is tested after executing the statements of the loop

**Lab Tasks**

**Task A**

Given two variables, we need to swap their values.

e.g Before execution of program: x=5 and y=10

After execution of program: x=10 and y=5

1. Write pseudocode
2. Draw flowchart
3. Write C++ program

Read a and b

Read num

Display a and b before swapping

Assign the value of a to num(num=a)

Assign the value of a to b (a=b)

And the store the value of b into num(b=num)

Display the value of a and b after swapping

| int swappingOfNumbers(int &num1, int &num2)  {  int num;  num=num1; //storing the value of num in num1  num1=num2; //storing the value of num1 in num2  num2=num; //storing the value of num2 in num  cout<<"\nafter swapping"<<endl; //displaying after swapping  cout<<"\nnum1 = "<<num1<<" num2 = "<<num2<<endl; //displaying swapped values  return 0;  } |
| --- |

**Task B**

Generate Fibonacci Sequence up to a certain number using do while loop.

| The Fibonacci Sequence is such that each number is the sum of the two preceding ones, starting from 0 and 1.  0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...  The next number is found by adding up the two numbers before it:   * the 2 is found by adding the two numbers before it (1+1), * the 3 is found by adding the two numbers before it (1+2), * the 5 is (2+3), * and so on! |
| --- |

1. Write pseudocode
2. Draw flowchart
3. Write C++ program

| PSEUDO CODE:  Read n  Read a=0  Read b=1  Read c  Take the range of sequence  While (a<=n)  c=a+b  a=b  b=c  C++  int generateSequence()  {  int n, a=0, b=1, c; //initializing n ,a,b,c  // take input  cout << "Enter Range: "; //entering range at which we want series  cin >> n;  cout << "Fibonacci Series is: " << endl; //displaying series  do  {  cout << a << " ";  c=a+b; //adding a and b and storing it in c  a=b; //storing a in b  b=c; //storing b in c  }  while ( a <= n ); //condition of while that is a is less than equal to n  } |
| --- |

|  |
| --- |

#### **Assessment Rubric for Lab**

**Method for assessment:**

Lab reports and instructor observation during lab sessions. Outcome assessed:

a. Ability to conduct experiments, as well as to analyze and interpret data (P) b. Ability to function on multi-disciplinary teams (A)

c. Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (P)

| Performance metric | Mapping (task no. and description) | | Max marks | Exceeds expectation | Meets expectation | Does not meet expectation | Obtained marks |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Realization of experiment (a) | 1 | Functionality | 40 | Executes without errors excellent user prompts, good use of symbols, spacing in output. Through testing has been completed (35-40) | Executes without errors, user prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed (20-34) | Does not execute due to syntax errors, runtime errors, user prompts are misleading or non-existent. No testing has been completed (0-19) |  |
| 2. Teamwork (b) | 1 | Group Performance | 5 | Actively engages and cooperates with other group member(s) in effective manner (4-5) | Cooperates with other group member(s) in a reasonable manner but conduct can be improved (2-3) | Distracts or discourages other group members from conducting the experiment (0-1) |  |
| 3. Conducting experiment (a, c) | 1 | On Spot Changes | 10 | Able to make changes (8-10) | Partially able to make changes (5-7) | Unable to make changes (0-4) |  |
| 2 | Viva | 10 | Answered all questions (8-10) | Few incorrect answers (5-7) | Unable to answer all questions (0-4) |  |
| 4. Laboratory safety and disciplinary rules (a) | 1 | Code commenting | 5 | Observes lab safety rules; handles the equipment and parts with care and adheres to the lab disciplinary guidelines aptly (4-5) | Generally observes safety rules and disciplinary guidelines with minor lapses (2-3) | Disregards lab safety and disciplinary rules (0-1) |  |
| 5. Data collection (c) | 1 | Code Structure | 5 | Excellent use of white space, creatively organized work, excellent use of variables and constants, correct identifiers for constants, No line-wrap (4-5) | Includes name, and assignment, white space makes the program fairly easy to read. Title, organized work, good use of variables (2-3) | Poor use of white space (indentation, blank lines) making code hard to read, disorganized and messy (0-1) |  |
| 6. Data analysis (a, c) | 1 | Algorithm | 20 | Solution is efficient, easy to understand, and maintain (15-20) | A logical solution that is easy to follow but it is not the most efficient (6-14) | A difficult and inefficient solution (0-5) |  |
| 7. Computer use (c) | 1 | Documentation | 5 | Timely documented (4-5) | Late documented (2-3) | Not documented (0-1) |  |
|  | Max Marks (total): | | 100 | Obtained Marks (total): | | |  |

Lab Engineer Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_