Faculty of Computing

CS110: Fundamentals of Computer Programming

Class: BESE-16B

Lab 04: Repetition Statements-Loops

CLO 2	**Solve** given real-world problem by applying appropriate programming concepts and techniques
CLO 3	**Build** a program and associated documentation using appropriate IDE and supplementary tools

Date: 30th September, 2025

Time: 2:00pm-5:00pm

Instructor: Dr. Momina Moetesum
Lab Engineer: Mr. Nadeem Nawaz

Introduction:

This lab focuses on *repetition statements* in C++, a fundamental programming construct that enables repetition of tasks within programs. Students will learn how to use while, do-while, and for-loops to control the flow of execution based on specific conditions. By working through guided exercises, students will practice applying these constructs to solve practical problems. The lab also emphasizes hands-on use of the Visual Studio IDE for compiling, running, and debugging programs, ensuring students gain both conceptual understanding and practical programming experience.

Learning Objectives:

After completing this section, you will be able to:

- **Apply** repetition constructs to solve real-world problems by implementing C++ programs.
- **Build** simple C++ programs and report outcomes.

Tools/Software Requirement:

- Microsoft Visual Studio (any version)
- C++ Compiler (integrated within Visual Studio)
- Word Processor (MS Word or equivalent for compiling deliverables)

Task 1 [CLO 2]: Number Guessing Game

Use while loop to write a C++ program that prompts the user to guess a number between 1 to 10. You must store the secret number in a variable and use that variable throughout for checks and count the number of tries it takes the user to guess it. The user will try to guess the secret number until they get it right. That means it will keep looping as long as the guess is different from the secret number.

Example outputs are provided for reference.

Sample output

```
I have chosen a number between 1 and 10. Try to guess it.
Your guess: 5
That is incorrect.  Guess again.
Your guess: 4
That is incorrect.  Guess again.
Your guess: 8
That is incorrect.  Guess again.
Your guess: 6
That's right! You guessed it.
It only took you 4 tries.
```

Task 2 [CLO 2]: Computing Harmonic Mean

The harmonic mean is another way of calculating a mean for a set of numbers. The harmonic mean of a set of **N** numbers is given by the equation:

$$\text{harmonic\_mean} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_N}}$$

Use a do-while loop to write a C++ program that prompts the user to enter the value of N (e.g. N = 10) and calculates the harmonic mean of the set of N numbers. Test your program's output for N=10 and N=4.

Task 3 [CLO 3]:

The factorial of an integer N, written N! is simply the product of all the integers from 1 to n. For example, 5! = 1*2*3*4*5 = 120. **Write** a C++ program usinf for-loop to calculate the factorial of an integer N taken input from the user.

Task 4 [CLO 3]:

Write a C++ program in your IDE. The program should prompt the user to enter the number of items and each item's name and price, and finally displays the name and price of the item with the lowest price. Use **any loop of your choice** to attempt this task.

Deliverables:

Compile a single Word document as displayed in solution/answer part and submit this Word file on LMS.

Lab Rubrics:

Your Lab 4 will be graded out of 5 for each rubric according to the following rubrics.

Lab Rubrics for Lab 4					
Sr. No.	**Assessment**	**Unacceptable (0 Marks)**	**Does Not Meet Expectations (1/2 Marks)**	**Meets Expectations (3/4 Marks)**	**Exceeds Expectations (5 Marks)**
1	**Application of Programming Concepts (CLO2, PLO3)**	The student did not submit any work. OR The student plagiarized the solution and/or used unfair means.	The student is unable to apply the appropriate programming concepts to solve the given problem thus resulting in an incomplete or ineffective solution. The program flow is messy and incomprehensible. Codes are non-modular and cannot be reused.	The student requires some guidance to apply the appropriate programming concepts to solve the given problem. The program flow requires minor improvements. Codes are semi-modular and semi-reusable.	The student demonstrates a clear ability to apply the appropriate programming concepts to solve the given problem. The program flow is adequate. Codes are modular, reusable, and easily readable.
2	**Software Tool Usage (CLO3-PLO5)**		The student demonstrates a lack of understanding of tool usage. Implementation has syntax/semantic/runtime errors, and the student is unable to debug and correct the errors. The code has inadequate comments and variable names and does not adhere to the coding standards. No Error handling has been performed. Documentation is poorly structured.	The student demonstrates some understanding of tool usage. The codes are correct in terms of their syntax, however, the program output is not always correct in all test cases. The code has limited comments and inconsistent variable names and may not adhere to the coding standards. Some Error handling has been performed. Documentation is adequately structured.	The student demonstrates a good understanding of tool usage. Furthermore, his/her coding is complete and functional, and the program output is correct in all test cases. The code has sufficient comments and consistent variable names and reasonably adhere to the coding standards. Adequate Error handling has been performed. Documentation is well structured.