Faculty of Computing

CS110: Fundamentals of Computer Programming

Class: BESE-16B

Lab 14: User Defined Variables using Structures

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: 16th December 2025

Time: 2:00 pm-5:00 pm

Instructor: Dr. Momina Moetesum

Lab Engineer: Mr. Nadeem Nawaz

Lab 14: User Defined Variables using Structures

Learning Objectives:

After completing this section, you will be able to:

- Define and use structures to represent real-world objects in C++.
- Access and manipulate members of structures.
- Use arrays of structures to handle multiple objects.
- Pass structures to functions for processing and problem-solving.

Lab Tasks

Task 1[CLO2]

Apply the concept of structures in C++ by defining a structure to represent a point in two-dimensional space. Use this structure to create and initialize two points, compute the Euclidean distance between them using their coordinate values, and display the coordinates of both points along with the calculated distance. Ensure that appropriate header files are included in the program.

Task 2: [CLO 2]

Use structures to create a Clock object to represent time. Prompt the user to enter hours, minutes, and seconds, then display the time in HH:MM:SS format. Validate that hours are between 0-23, minutes 0-59, and seconds 0-59. If invalid, display an error message. Here is a sample run:

Enter hours: 14
Enter minutes: 30
Enter seconds: 45
Time: 14:30:45
Enter hours: 25
Invalid hours!

Task 3: [CLO 3]

Write a program that uses a structure to represent a Student object with members: name (string), rollNumber (int), and marks (float).
Create an array of 3 Student structures. Prompt the user to input data for each student, then calculate and display the average marks of the class. Here is a sample run:

Enter name for student 1: Alice
Enter roll number: 101
Enter marks: 85.5
Enter name for student 2: Bob
Enter roll number: 102
Enter marks: 92.0
Enter name for student 3: Charlie
Enter roll number: 103
Enter marks: 78.5
Class average marks: 85.3333

Task 4: [CLO 3]

Structures in C++ can be used to model complex real-world entities by grouping related data.

```
struct Book {
char title[50];
char author[30];
int year;
float price;
};
```

You can process structures using functions that take them as parameters or return them. Some example operations include initializing, displaying, or comparing structures. A simple library management system tracks books using ISBN as a unique identifier. However, for this task, focus on validating book entries.

Write a program that defines a structure for a Book object as above. Create a function bool isValidBook(const Book& b) that checks if the book's year is between 1900 and 2023, price is positive, and title is not empty. In the main function, prompt the user to enter details for one book, then use the function to validate and display "Valid Book" or "Invalid Book". If valid, display the book details.

Deliverables:

Compile a single Word document with codes for each question and screenshots of the outputs and submit this Word file on LMS.

Lab Rubrics

Your Lab 14 will be graded out of 5 for each rubric according to the following rubrics. Grades for CLO3 will be shared at different intervals during the semester.

Lab Rubrics for Lab 14 (User Defined Variables using Structures)

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.