Week #4

Objective:
The objective of this lab is to introduce students to the concept of structures in C++ and familiarize them with their usage. By the end of this lab, students should be able to understand how to define, initialize, and manipulate structures, as well as use them effectively in your programs.

Introduction to Structures:
In C++, a structure is a user-defined data type that allows you to group together variables of different data types under a single name. Structures are used to represent a collection of related data items, making it easier to manage and manipulate complex data structures in your programs.

Key Concepts:

Definition of Structures: Structures are defined using the **struct** keyword followed by the structure name and a set of members enclosed in curly braces.

```cpp
struct StructureName {
    dataType member1;
    dataType member2;
    // More members...
};
```

```cpp
struct Person {
    string name;
    int age;
    float height;
};
```

Structure Members: Each variable inside a structure is called a member or field, and they can have different data types.

Accessing Structure Members: Members of a structure can be accessed using the dot (.) operator.

```cpp
Person p1;
p1.name = "John";
p1.age = 30;
p1.height = 6.1;
```

Nested Structures: Structures can be nested within other structures, allowing for the representation of hierarchical data.

```cpp
struct Address {
    string street;
    string city;
};

struct Employee {
    string name;
    int age;
    Address address;
};
```

Arrays of Structures: Arrays of structures enable the storage of multiple instances of the same type of data. You can create arrays of structures to store multiple instances of the same type of data. Each element of the array is a structure variable. Arrays of structures are useful for managing collections of related data.

Pointers to Structures: Pointers can be used to dynamically access and manipulate structures. Pointers can be used to access structures dynamically and efficiently. Pointer notation (->) is used to access members of a structure through a pointer.

Use Cases of Structures:
Structures are commonly used to represent real-world entities such as employees, students, books, etc.
They are useful for organizing and managing data in a structured manner.
Structures facilitate code readability, maintainability, and modularity.

By practicing these concepts through hands-on exercises and challenges, the students will enhance their problem-solving skills and develop a deeper understanding of how to write efficient and effective C++ code. So, let's get started and embark on this journey to master the art of programming logic!

Exercise:

Q.1) Define a structure named Person with members name and age. Create a variable of this structure type, initialize it, and display its contents.

Q.2) Define a structure named Rectangle with members length and width. Create a variable of this structure type, input values for length and width, calculate the area, and display it.

Q3) Define a structure named Number with a single member num. Create two variables of this structure type, input values for them, swap their values using a function, and display the swapped values.

Q4 Define a structure named Student with members rollNumber, name, age, and marks. Create an array of 3 Student structures, input data for each student, display their details, and find the student with the highest marks.

Q5) Define a structure named Employee with members employeeID, name, department, and salary. Create an array of 5 Employee structures, input data for each employee, display their details, and calculate the average salary of all employees.

Q6) Define a structure named Book with members title, author, genre, and yearOfPublication. Create an array of 3 Book structures, input data for each book, display their details, and find the book with the earliest publication year.

Q7) Define a structure named Product with members productName, price, and quantity. Create a variable of this structure type, input data for the product, calculate the total cost, and display it.