# C++ OOP and Data Structures

**Instructions:**

* Answer all questions.
* Write clear and commented code where necessary.
* Duration: 2 hours.

## Problem 1: Class with Array and Loop

Create a class `Student` with:  
1. A private attribute `marks` (array of 5 integers).  
2. A method `setMarks()` to take input for the 5 marks.  
3. A method `calculateAverage()` to calculate and return the average of the marks using a loop.  
4. A method `displayMarks()` to print all 5 marks.  
  
Write a `main()` function to create a `Student` object, input marks, calculate the average, and display the marks and average.

## Problem 2: Queue Operations in a Class

Create a class QueueOperations with:

1. A private attribute queue<int> data.
2. Methods:
   * enqueue(int n) to add an element to the queue.
   * dequeue() to remove the front element.
   * displayQueue() to print all elements in the queue.

Write a main() function to:

1. Enqueue 5 integers into the queue.
2. Dequeue the first 2 elements.
3. Display the remaining elements.

## Problem 3: Stack and Encapsulation

Create a class `StackOperations` with:  
1. A private attribute `stack<int> data`.  
2. Methods to perform the following stack operations:  
 - `push(int n)` to add an element to the stack.  
 - `pop()` to remove the top element.  
 - `displayStack()` to print all elements of the stack (use a loop).  
  
Write a `main()` function to demonstrate:  
1. Pushing 5 integers onto the stack.  
2. Popping the top 2 elements.  
3. Displaying the remaining elements.

## Problem 4: Matrix Operations

Create a class Matrix with:

1. A private attribute vector<vector<int>> mat to store a 2D matrix.
2. Methods:
   * setMatrix(int rows, int cols) to take input for a matrix of given dimensions.
   * transpose() to calculate and return the transpose of the matrix.
   * displayMatrix() to print the matrix.

Write a main() function to:

1. Create a Matrix object.
2. Input a 2D matrix.
3. Display the original and transposed matrices.

## Problem 5: Sorting and Searching in OOP

Create a class `DataProcessor` with:  
1. A private attribute `vector<int> data`.  
2. A method `addData(int n)` to add elements to the vector.  
3. A method `sortData()` to sort the vector in ascending order using the `sort()` function.  
4. A method `searchData(int n)` to perform a linear search for an element in the vector.  
  
Write a `main()` function to:  
1. Add 10 integers to the vector.  
2. Sort the vector and display the sorted elements.  
3. Search for a specific number and display whether it is found or not.

## Problem 6: Shopping Cart System

Create a class ShoppingCart with:

1. A private attribute map<string, int> items where the key is the item name, and the value is the quantity.

2. Methods:

* + addItem(string itemName, int quantity) to add or update the quantity of an item in the cart.
  + removeItem(string itemName) to remove an item from the cart if it exists.
  + displayCart() to display all items and their quantities in the cart.
  + Write a main() function to:

1. Add 5 items to the shopping cart.

2. Remove 1 item from the cart.

3. Display the updated cart items.