Practical Exam: Data Structures and Algorithms with C++

Total Marks: 100 Marks
Time Duration: 2 Hours
Instructions: Solve all questions. Each question carries 10 marks. Provide proper code implementation and comments.

Question 1: Reverse an Array
Problem: Write a function to reverse an array of integers.

Input Example:
Input: [1, 2, 3, 4, 5]

Output Example:
Output: [5, 4, 3, 2, 1]

Question 2: Find the Middle Element of a Linked List
Problem: Implement a function to find the middle element of a singly linked list.

Input Example:
Input: 1 -> 2 -> 3 -> 4 -> 5

Output Example:
Output: 3

Question 3: Implement Stack using Array
Problem: Create a stack using an array and perform the following operations: Push, Pop, and Display.

Input Example:
Push: 10, 20, 30
Pop: 1 (Remove the top element)

Output Example:
Stack after operations: [10, 20]

Question 4: CRUD Operations using Class and Object
Problem: Implement a class `Student` with attributes `id`, `name`, and `marks`. Write functions to perform CRUD (Create, Read, Update, Delete) operations on `Student` objects.

Input Example:
Add: {id: 1, name: "John", marks: 85}
Update: id = 1, marks = 90

Output Example:
Updated Details: {id: 1, name: "John", marks: 90}

Question 5: Use of Constructors in C++

Problem: Implement a class `Rectangle` with attributes `length` and `breadth`. Use a parameterized constructor to initialize the attributes and calculate the area of the rectangle.

Input Example:
Length: 5, Breadth: 10

Output Example:
Area: 50

Question 6: Sort an Array using Merge Sort

Problem: Write a program to sort an array using the merge sort algorithm.

Input Example:
Input: [38, 27, 43, 3, 9]

Output Example:
Output: [3, 9, 27, 38, 43]

Question 7: Working with Vectors in C++

Problem: Write a program to demonstrate the use of vectors. Perform the following operations: Add elements, access elements, modify elements, and delete an element from the vector.

Input Example:
Add: [10, 20, 30, 40], Remove: 30

Output Example:
Updated Vector: [10, 20, 40]

Question 8: Implement Queue using Linked List

Problem: Create a queue using a linked list and perform the following operations: Enqueue, Dequeue, and Display.

Input Example:
Enqueue: 5, 10, 15
Dequeue: 1

Output Example:
Queue after operations: [10, 15]

Question 9: Create a Class for Matrix Operations

Problem: Implement a class `Matrix` with member functions for addition, subtraction, and multiplication of matrices. Use proper constructors to initialize the matrices.

Input Example:
Matrix A:
1 2
3 4
Matrix B:
5 6
7 8

Output Example:
Matrix Addition Result:
6 8
10 12

Question 10: Create a To-Do List Application using Class

Problem: Implement a class `ToDoList` that supports adding tasks, viewing all tasks, marking a task as completed, and deleting tasks. Use a vector to store tasks.

Input Example:
Add: "Complete assignment", "Attend meeting"
Mark Complete: 1

Output Example:
Remaining Tasks: "Attend meeting"