|  |  |  |
| --- | --- | --- |
|  | **American International University – Bangladesh (AIUB) Faculty of Science and Technology (FST) Department of Computer Science (CS)** | Course: Data Structure Lab  Final Term Assignment Spring 24-25 |

### Instructions

Use **proper C++ concepts** mentioned in **Function and System Requirements** section in the problem scenario. Use proper coding structure that should highlight your coding knowledge.

### What You Will Submit

You have to create a .cpp file and the file name should be your ID. **(Example: 11-12345-2.cpp)**

### Where To Submit

Submit only the .cpp file **(no zip file required)** in the assignment section named **(DS Lab Final Term Assignment Submission)** on MS Teams. After submitting the file, you must press the **“Turn In”**/ **”Hand In”** button.

**Submission Deadline: 20 June, 2025 at 11:59 PM**

### What You Should Not Do

To maintain academic integrity and ensure fair evaluation, students are strictly advised against any form of plagiarism. This includes copying code or solutions from online sources such as ChatGPT, other AI tools, coding websites, or from friends. Assignments must reflect your own understanding, logic, and effort. You are also advised to follow the deadline strictly. Any violation of these rules will result in deducting marks.

# University Enrollment and Exam Management System

## Problem Scenario

You are tasked with building a simulation to track student numbers, course registration queues, and exam scores using numerical values only. The system should use the following data structures:

**1. Student Admission System (Doubly Linked List)**

Maintain a list of admission numbers using a **doubly linked list**.

**Initial Admission Numbers:**  
12 -> 25 -> 38 -> 41

**Operations to perform:**

* Add 56 to the **end** of the list.
* Delete the number 25.
* Display the list **forward**.
* Display the list **backward**.

**2. Course Registration System (Queue)**

Simulate a **course registration queue** using a simple queue structure. Each entry is a number representing a registration.

**Initial Queue Entries (in order):**  
101, 102, 103

**Operations to perform:**

* Enqueue 104
* Enqueue 105
* Dequeue **once**
* Display the **current queue**

**3. Exam Result Management (Binary Search Tree – BST)**

Manage exam results using a **Binary Search Tree (BST)**. Each node contains a number (score).

**Initial Scores to Insert (in order):**  
60 (root), 30, 10, 40, 20, 80, 100, 70, 90

**Operations to perform:**

* Display results using **Inorder Traversal**
* Display results using **Preorder Traversal**
* Display results using **Postorder Traversal**

# Sample Input Commands in the Main Function

# AddStudent(12);

# AddStudent(25);

# AddStudent(38);

# AddStudent(41);

# AddStudent(56);

# DeleteStudent(25);

# DisplayStudentsForward();

# DisplayStudentsBackward();

# Enqueue(101);

# Enqueue(102);

# Enqueue(103);

# Enqueue(104);

# Enqueue(105);

# Dequeue();

# DisplayCourseQueue();

# InsertScore(60);

# InsertScore(30);

# InsertScore(10);

# InsertScore(40);

# InsertScore(20);

# InsertScore(80);

# InsertScore(100);

# InsertScore(70);

# InsertScore(90);

# DisplayInorder();

# DisplayPreorder();

# DisplayPostorder();

# Sample Outputs based on the Input Commands

Student Added: 12

Student Added: 25

Student Added: 38

Student Added: 41

Student Added: 56

Student Deleted: 25

**Student List (Forward):**

12 → 38 → 41 → 56

**Student List (Backward):**

56 → 41 → 38 → 12

Course Enqueued: 101

Course Enqueued: 102

Course Enqueued: 103

Course Enqueued: 104

Course Enqueued: 105

Course Dequeued: 101

**Course Registration Queue:**

102 → 103 → 104 → 105

Score Inserted: 60

Score Inserted: 30

Score Inserted: 10

Score Inserted: 40

Score Inserted: 20

Score Inserted: 80

Score Inserted: 100

Score Inserted: 70

Score Inserted: 90

**Inorder Traversal:**

10 20 30 40 60 70 80 90 100

**Preorder Traversal:**

60 30 10 20 40 80 70 100 90

**Postorder Traversal:**

20 10 40 30 70 90 100 80 60