CS214 – Student Record Management System

Submission: Saturday, 19/04/2025, 11:59 PM

Project Level: Intermediate
Group Size: Maximum 3 Students
Total Marks: 10
Submission Platform: Blackboard

1. Project Overview

In this project, you will develop a **Student Record Management System** using multiple data structures. The system will allow users (e.g., administrators or teachers) to manage student records efficiently. The goal is to apply your knowledge of **arrays, linked lists, stacks, and queues** to build a functional application.

2. Functional Requirements

Your system should support the following operations:

Basic Operations

- **Add a student record** (ID, name, GPA, department, etc.)
- **Delete a student record** using ID
- **Update a student record** (e.g., change GPA)
- **Search for a student** by ID
- **Display all student records**

Additional Features (Using Data Structures)

- **Undo last operation** (Using Stacks)
- **Handle student requests (help desk)** on a first-come, first-served basis (Using Queues)

3. Data Structures & Justification

Each data structure plays a role in optimizing specific operations:

Data Structure	Purpose
Array	Store student records in a fixed-size structure (initial implementation)
Linked List	Allow dynamic insertion and deletion of students
Stack	Enable "Undo" functionality for the last operation

Data Structure	Purpose
Queue	Manage student requests in First-Come, First-Served order

---

4. Project Deliverables

A. Code Implementation (Uploaded to Blackboard)

- Your project should be implemented in **C++**.
- Provide a **command-line interface (CLI)** for easy interaction.
- Follow **proper coding standards** (indentation, comments, meaningful variable names).
- Ensure **error handling** (e.g., handle invalid inputs gracefully).

B. Final Report (Uploaded to Blackboard)

Your report should include:

1. **Introduction** – Explain the project's purpose and features.
2. **Data Structures Used** – Justify your choice of data structures.
3. **Implementation Details** – Explain your approach and key functions.
4. **Challenges & Learnings** – Describe the problems you faced and how you solved them.
5. **Conclusion** – Summarize the project and future improvements.

C. Presentation (5 Minutes Per Group)

- Demonstrate your project's functionality.
- Explain your data structure choices.
- Discuss challenges and how you overcame them.

---

5. Grading Criteria (15 Marks Total)

Criteria	Marks	Description
Implementation of required data structures	**5 Marks**	Correct use of arrays, linked lists, stacks, and queues.
Code efficiency and correctness	**2 Marks**	Logical, error-free, and well-optimized implementation.
Report quality and clarity	**1 Marks**	Well-structured report covering all sections.
Teamwork & problem-solving approach	**2 Marks**	Effective collaboration, innovation, and problem-solving.

---

6. Submission Guidelines

- The **final code and report** must be submitted **via Blackboard** before the deadline.
- **Late submissions will result in mark deductions** unless valid reasons are provided.

---

7. Notes & Suggestions

- **Divide tasks among team members** to work efficiently.
- **Debug and test your code thoroughly before submission.**
- **Use proper formatting in your report for clarity.**

---

8. Late Submission Policy

- **Extensions may be granted only if requested before the due date.**
- **Without prior approval, late submissions will incur a penalty of 5% per day (maximum three days).**
- **After three days, no submission will be accepted, and a zero grade will be recorded.**
- **Documentation (e.g., medical notes) may be required for exemption.**

Good luck with your project!