

# TASK 1

## Project Report

### Simple Student Management System in C++

---

#### 1. Introduction

This project implements a basic console-based Student Management System using C++. It allows users to add, display, and delete student records. All data can be saved to and loaded from a file, so that student records are not lost between program runs. The project focuses on practicing the use of **structs**, **arrays**, **file input/output (I/O)**, **pointers**, and **menu-driven programming**.

---

#### 2. Project Features

- **Add Student:** Input a student's ID, name, and grade.
  - **Display Students:** Show a list of all students currently stored.
  - **Delete Student:** Remove a student by entering their ID.
  - **Save Records:** Write the current list of students to a binary file (students.txt).
  - **Load Records:** Read existing student data from the file at program start.
  - **Menu-Driven:** The user interacts with a clear menu for easy operation.
- 

#### 3. Tools and Concepts Used

- **Structs:** Used to create a Student structure containing ID, name, and grade.
  - **Arrays:** A fixed-size array is used to store multiple student records.
  - **File Handling:**
    - ofstream and ifstream are used to save and load data in binary format.
  - **Functions:** Separate functions handle each operation (add, delete, save, load, display).
  - **Pointers:** Basic use when dealing with file I/O (binary read/write operations).
- 

#### 4. Program Flow

1. On startup, the program tries to **load existing records** from a file.
2. It then **displays a menu** with choices (add, display, delete, save, exit).

3. The user can perform multiple actions.
4. Upon exiting, the program automatically **saves** any changes made during the session.

---

## 5. Time Complexity

| Operation | Time Complexity |
|-----------|-----------------|
|-----------|-----------------|

|                  |        |
|------------------|--------|
| Adding a student | $O(1)$ |
|------------------|--------|

|                     |        |
|---------------------|--------|
| Displaying students | $O(n)$ |
|---------------------|--------|

|                    |        |
|--------------------|--------|
| Deleting a student | $O(n)$ |
|--------------------|--------|

|                |        |
|----------------|--------|
| Saving to file | $O(n)$ |
|----------------|--------|

|                   |        |
|-------------------|--------|
| Loading from file | $O(n)$ |
|-------------------|--------|

where **n** = number of students.

---

## 6. Conclusion

Through this project, I practiced working with structs, arrays, file handling, and functions in C++. I learned how to implement a simple CRUD (Create, Read, Update, Delete) system and how to persist data using file operations. This project also helped strengthen my understanding of organizing code into manageable functions and creating user-friendly, menu-driven applications.