DESIGNING A STUDENT MARKS MANAGEMENT SYSTEM IN C++ WITH FILE HANDLING

Executive Summary

This case study demonstrates how to build a console-based Student Marks Management System using C++ and file handling techniques. It guides students through the core logic behind storing, updating, retrieving, and deleting student records from a file. The project uses structures, menus, basic file I/O, and conditional logic, helping students strengthen foundational C++ concepts and understand real-world file operations in academic systems.

---

1. Introduction

Many college and university lab assignments require students to build small systems for academic record management. These typically involve a menu-driven program and some form of persistent storage. This project solves the problem using C++ structures and text/binary file operations, aiming to provide students with clear logic and error-handling strategies.

---

2. Problem Definition

Design a simple system to manage:

- Adding new student records (Name, Roll No., Marks in 3 subjects)
- Displaying all records
- Searching by Roll Number
- Modifying an existing record
- Deleting a record

---

3. System Requirements

- **Language**: C++
- **Concepts Covered**: Structures, File Handling (fstream), Menus, Conditional Logic

- **Storage**: Text file students.txt or binary file students.dat
- **I/O**: Standard CLI using cin, cout

---

4. Core Structure and Logic

cpp

CopyEdit

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

struct Student {
    int roll;
    string name;
    float marks1, marks2, marks3;
};

void addStudent() {
    Student s;
    ofstream fout("students.txt", ios::app);
    cout << "Enter Roll No, Name, Marks in 3 subjects:\n";
    cin >> s.roll >> s.name >> s.marks1 >> s.marks2 >> s.marks3;
    fout << s.roll << " " << s.name << " " << s.marks1 << " " << s.marks2 << " " << s.marks3 <<
"\n";
    fout.close();
}
```

Explanation:

- struct Student stores individual records.
- ofstream in append mode writes new records to file.
- Basic CLI input is used; real-time error checks can be added.

---

5. File Read and Display Logic

cpp

CopyEdit

```
void displayAll() {
    Student s;
    ifstream fin("students.txt");
    cout << "Roll\tName\tMarks1\tMarks2\tMarks3\n";
    while (fin >> s.roll >> s.name >> s.marks1 >> s.marks2 >> s.marks3) {
        cout << s.roll << "\t" << s.name << "\t" << s.marks1 << "\t" << s.marks2 << "\t" << s.marks3 << "\n";
    }
    fin.close();
}
```

Other functions (search, delete, update) follow the pattern:

- Read all records
- Copy or modify them to a temporary file
- Replace the original file with the updated one

---

6. Menu and Main Loop

cpp

CopyEdit

```
int main() {
```

```
    int choice;
    do {
        cout << "\n1. Add\n2. Display\n3. Exit\nEnter choice: ";
        cin >> choice;
        switch (choice) {
            case 1: addStudent(); break;
            case 2: displayAll(); break;
            case 3: break;
            default: cout << "Invalid option";
        }
    } while (choice != 3);
    return 0;
}
```

---

7. Evaluation and Common Pitfalls

Beginner Errors Solved:

- Forgetting to close file streams
- Overwriting files when not using ios::app
- Reading from files with wrong formatting
- Mixing text and binary modes

Learning Outcomes:

- Understanding how data is written and read in files
- Structuring CLI systems using menus
- Managing persistent storage without databases

---

8. Suggested Enhancements

- Use binary files with read() and write() for faster access
- Implement average calculation and grade assignment logic
- Add password protection for access
- Use file locking or backup logic to prevent data corruption

---

9. Conclusion

This case study provides a step-by-step logic explanation of building a student marks system using C++. It offers strong fundamentals in struct handling, file operations, and user interface logic—all applicable to mini-projects and lab assignments.

---

10. References

- C++ Primer (Lippman, 5th Edition)
- GeeksforGeeks: File Handling in C++
- Cplusplus.com Documentation for fstream