Practical no. 11

```
/*
Title:Department maintains student information. The file contains roll
number, name, division and address. Allow users to add, delete information
about students. Display information of a particular employee. If the record
of the student does not exist an appropriate message is displayed. If it is,
then the system displays the student details. Use a sequential file to
maintain the data.
*/


#include <iostream>
#include <fstream>
#include <cstring>
using namespace std;
struct Student {
    int rollNumber;
    char name[50];
    char division;
    char address[100];
};
void addStudent() {
    ofstream outFile("students.dat", ios::binary | ios::app);
    if (!outFile) {
        cout << "Error opening file. Please try again." << endl;
        return;
    }

    Student student;

    cout << "Enter Roll Number: ";
    cin >> student.rollNumber;

    cout << "Enter Name: ";
    cin.ignore();
    cin.getline(student.name, 50);
    cout << "Enter Division: ";
    cin >> student.division;

    cout << "Enter Address: ";
    cin.ignore();
    cin.getline(student.address, 100);

    outFile.write(reinterpret_cast<char*>(&student), sizeof(Student));
    outFile.close();

    cout << "Student information added successfully." << endl;
}

void deleteStudent() {
    int rollNumber;
    cout << "Enter the Roll Number of the student you want to delete: ";
    cin >> rollNumber;

    ifstream inFile("students.dat", ios::binary);
    if (!inFile) {
        cout << "Error opening file. Please try again." << endl;
        return;
    }

    ofstream tempFile("temp.dat", ios::binary);
    if (!tempFile) {
        cout << "Error creating temporary file. Please try again." << endl;
        inFile.close();
        return;
    }

    Student student;
    bool found = false;

    while (inFile.read(reinterpret_cast<char*>(&student), sizeof(Student))) {
        if (student.rollNumber != rollNumber) {
            tempFile.write(reinterpret_cast<char*>(&student), sizeof(Student));
        } else {
            found = true;
        }
    }

    inFile.close();
    tempFile.close();

    remove("students.dat");
    rename("temp.dat", "students.dat");

    if (found) {
        cout << "Student information deleted successfully." << endl;
    } else {
        cout << "Student not found. Please check the Roll Number." << endl;
    }
}

void displayStudent() {
    int rollNumber;
    cout << "Enter the Roll Number of the student you want to display: ";
    cin >> rollNumber;

    ifstream inFile("students.dat", ios::binary);
    if (!inFile) {
        cout << "Error opening file. Please try again." << endl;
        return;
    }

    Student student;
    bool found = false;

    while (inFile.read(reinterpret_cast<char*>(&student), sizeof(Student))) {
        if (student.rollNumber == rollNumber) {
            cout << "Roll Number: " << student.rollNumber << endl;
            cout << "Name: " << student.name << endl;
            cout << "Division: " << student.division << endl;
            cout << "Address: " << student.address << endl;
            found = true;
            break;
        }
    }

    inFile.close();

    if (!found) {
        cout << "Student not found. Please check the Roll Number." << endl;
    }
}

int main() {
    int choice;
    while (true) {
        cout << "----------------------------" << endl;
        cout << "Student Information Database" << endl;
        cout << "----------------------------" << endl;
        cout << "1. Add Student" << endl;
        cout << "2. Delete Student" << endl;
        cout << "3. Display Student" << endl;
        cout << "4. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice) {
          case 1:addStudent();
                    break;
          case 2:deleteStudent();
                    break;
          case 3:displayStudent();
                    break;
          case 4:
          cout << "Exiting the program.";
                return 0;
         default:
                cout << "Invalid choice. Please try again." << endl;
        }

        cout << endl;
    }

    return 0;
```

```
}
```

Output:

```
----------------------------
Student Information Database
----------------------------
1. Add Student
2. Delete Student
3. Display Student
4. Exit
Enter your choice: 1
Enter Roll Number: 40
Enter Name: Prafulla
Enter Division: B
Enter Address: Dholwad
Student information added successfully.

----------------------------
Student Information Database
----------------------------
1. Add Student
2. Delete Student
3. Display Student
4. Exit
Enter your choice: 3
Enter the Roll Number of the student you want to display: 40
Roll Number: 40
Name: Prafulla
Division: B
Address: Dholwad

----------------------------
Student Information Database
----------------------------
```

```
1. Add Student
2. Delete Student
3. Display Student
4. Exit
Enter your choice: 2
Enter the Roll Number of the student you want to delete: 40
Student information deleted successfully.

----------------------------
Student Information Database
----------------------------
1. Add Student
2. Delete Student
3. Display Student
4. Exit
Enter your choice: 4
Exiting the program.
```