**Lab Assignment 10**

**Name = ishwari Sahebrao Jeughale**

**Roll NO = 23527**

**batch = s2**

#include <iostream>

#include <fstream>

#include <string>

#include <cstring>

#include <iomanip>

using namespace std;

class Student

{

private:

    int rollNo;

    char name[50];

    char division;

    char address[100];

public:

    Student()

    {

        rollNo = 0;

        division = ' ';

        strcpy(name, "");

        strcpy(address, "");

    }

    void inputDetails()

    {

        cout << "\nEnter Roll Number: ";

        cin >> rollNo;

        cin.ignore();

        cout << "Enter Name: ";

        cin.getline(name, 50);

        cout << "Enter Division: ";

        cin >> division;

        cin.ignore();

        cout << "Enter Address: ";

        cin.getline(address, 100);

    }

    void displayDetails() const

    {

        cout << "\n----------------------------------------\n";

        cout << "Roll Number: " << rollNo << endl;

        cout << "Name: " << name << endl;

        cout << "Division: " << division << endl;

        cout << "Address: " << address << endl;

        cout << "----------------------------------------\n";

    }

    int getRollNo() const

    {

        return rollNo;

    }

};

class StudentDatabase

{

private:

    const char \*fileName;

public:

    StudentDatabase(const char \*file) : fileName(file) {}

    void addStudent()

    {

        Student student;

        ofstream file(fileName, ios::binary | ios::app);

        if (!file)

        {

            cout << "\nError opening file!\n";

            return;

        }

        student.inputDetails();

        if (searchStudent(student.getRollNo(), false))

        {

            cout << "\nStudent with Roll No. " << student.getRollNo() << " already exists!\n";

            file.close();

            return;

        }

        file.write(reinterpret\_cast<char \*>(&student), sizeof(Student));

        cout << "\nStudent record added successfully!\n";

        file.close();

    }

    void deleteStudent()

    {

        int rollNo;

        cout << "\nEnter Roll Number to delete: ";

        cin >> rollNo;

        ifstream fin(fileName, ios::binary);

        ofstream temp("temp.txt", ios::binary);

        if (!fin || !temp)

        {

            cout << "\nError opening file!\n";

            return;

        }

        Student student;

        bool found = false;

        while (fin.read(reinterpret\_cast<char \*>(&student), sizeof(Student)))

        {

            if (student.getRollNo() != rollNo)

            {

                temp.write(reinterpret\_cast<char \*>(&student), sizeof(Student));

            }

            else

            {

                found = true;

            }

        }

        fin.close();

        temp.close();

        if (found)

        {

            remove(fileName);

            rename("temp.txt", fileName);

            cout << "\nStudent record deleted successfully!\n";

        }

        else

        {

            remove("temp.txt");

            cout << "\nStudent record not found!\n";

        }

    }

    bool searchStudent(int rollNo, bool display = true)

    {

        ifstream file(fileName, ios::binary);

        if (!file)

        {

            cout << "\nError opening file!\n";

            return false;

        }

        Student student;

        bool found = false;

        while (file.read(reinterpret\_cast<char \*>(&student), sizeof(Student)))

        {

            if (student.getRollNo() == rollNo)

            {

                if (display)

                {

                    cout << "\nStudent found!";

                    student.displayDetails();

                }

                found = true;

                break;

            }

        }

        if (display && !found)

        {

            cout << "\nStudent with Roll No. " << rollNo << " not found!\n";

        }

        file.close();

        return found;

    }

    void displayAllStudents()

    {

        ifstream file(fileName, ios::binary);

        if (!file)

        {

            cout << "\nError opening file!\n";

            return;

        }

        Student student;

        bool found = false;

        cout << "\n============ All Student Records ============\n";

        while (file.read(reinterpret\_cast<char \*>(&student), sizeof(Student)))

        {

            student.displayDetails();

            found = true;

        }

        if (!found)

        {

            cout << "\nNo student records found!\n";

        }

        file.close();

    }

};

int main()

{

    StudentDatabase db("students.txt");

    int choice;

    do

    {

        cout << "\n====== Student Database Management System ======\n";

        cout << "1. Add Student\n";

        cout << "2. Delete Student\n";

        cout << "3. Search Student\n";

        cout << "4. Display All Students\n";

        cout << "5. Exit\n";

        cout << "Enter your choice: ";

        cin >> choice;

        switch (choice)

        {

        case 1:

            db.addStudent();

            break;

        case 2:

            db.deleteStudent();

            break;

        case 3:

        {

            int rollNo;

            cout << "\nEnter Roll Number to search: ";

            cin >> rollNo;

            db.searchStudent(rollNo);

            break;

        }

        case 4:

            db.displayAllStudents();

            break;

        case 5:

            cout << "\nExiting program...\n";

            break;

        default:

            cout << "\nInvalid choice! Please try again.\n";

            break;

        }

    } while (choice != 5);

    return 0;

}

**GitHub link:-**

[https://github.com/VedantKaulgekar/CollegeDSAL/blob/main/Assignment\_ HYPERLINK "https://github.com/VedantKaulgekar/CollegeDSAL/blob/main/Assignment\_10.cpp"10 HYPERLINK "https://github.com/VedantKaulgekar/CollegeDSAL/blob/main/Assignment\_10.cpp".cpp](https://github.com/VedantKaulgekar/CollegeDSAL/blob/main/Assignment_10.cpp)