#include <iostream>

#include <fstream>

using namespace std;

// Class to store student details

class student {

public:

int rollno; // Student's roll number

char name[50]; // Student's name

char add[50]; // Student's address

char div[5]; // Student's division

// Method to input student data

void setData() {

cout << "\nEnter Roll no: ";

cin >> rollno;

cout << "Enter Name: ";

cin.ignore(); // To ignore the leftover newline character from previous input

cin.getline(name, 50); // Input name

cout << "Address of student: ";

cin.getline(add, 50); // Input address

cout << "Division of student: ";

cin.getline(div, 5); // Input division

}

// Method to display student data

void showData() {

cout << "\nRoll no. of student: " << rollno;

cout << "\nName of the student: " << name;

cout << "\nAddress of student: " << add;

cout << "\nDivision of student: " << div << "\n";

}

// Method to return student's roll number for searching

int retrollno() {

return rollno;

}

};

// Function to write a student's record to the file

void write\_record() {

ofstream outfile("StudentData", ios::binary | ios::app); // Open file in binary append mode

student obj;

obj.setData(); // Get student data from the user

outfile.write((char\*)&obj, sizeof(obj)); // Write the object to file in binary format

outfile.close(); // Close the file after writing

}

// Function to display all student records from the file

void display\_record() {

ifstream infile("StudentData", ios::binary); // Open file in binary read mode

student obj;

while (infile.read((char\*)&obj, sizeof(obj))) { // Read each student record

obj.showData(); // Display the student's data

}

infile.close(); // Close the file after reading

}

// Function to search for a student record based on roll number

void search\_record(int n) {

bool flag = false;

ifstream infile("StudentData", ios::binary); // Open file in binary read mode

student obj;

while (infile.read((char\*)&obj, sizeof(obj))) { // Read each student record

if (obj.retrollno() == n) { // If roll number matches

flag = true;

obj.showData(); // Display the student's data

break; // Exit the loop after finding the record

}

}

infile.close(); // Close the file after reading

if (!flag) { // If record not found

cout << "\nRecord for roll no " << n << " not found!\n";

}

}

// Function to delete a student record based on roll number

void delete\_record(int n) {

student obj;

ifstream infile("StudentData", ios::binary); // Open the original file in binary read mode

ofstream outfile("StudentNewData", ios::binary); // Open a new file to store updated records

while (infile.read((char\*)&obj, sizeof(obj))) { // Read each student record

if (obj.retrollno() != n) { // If the roll number doesn't match, copy to new file

outfile.write((char\*)&obj, sizeof(obj));

}

}

infile.close(); // Close the original file

outfile.close(); // Close the new file

remove("StudentData"); // Remove the original file

rename("StudentNewData", "StudentData"); // Rename the new file to the original file name

cout << "\nRecord Deleted Successfully!!\n";

}

// Main function to perform file operations based on user choice

int main() {

int ch;

do {

// Menu for user input

cout << "\n\*File Operations\*"

<< "\n 1. Add Record."

<< "\n 2. Display Record."

<< "\n 3. Search a particular Record."

<< "\n 4. Delete particular Record."

<< "\n 5. Exit\n";

cout << "\nEnter your choice: ";

cin >> ch;

// Switch case to handle different operations based on user's choice

switch (ch) {

case 1: {

// Add records

cout << "\nEnter total no. of records you want to store in a file: ";

int n;

cin >> n;

for (int i = 0; i < n; i++) {

write\_record(); // Write each record to the file

}

break;

}

case 2:

cout << "\nList of records: \n";

display\_record(); // Display all records

break;

case 3: {

// Search for a particular record

cout << "\nEnter student roll no. whose record you want to search: ";

int s;

cin >> s;

search\_record(s); // Search and display the record

break;

}

case 4: {

// Delete a particular record

cout << "\nEnter student roll no. whose record you want to delete: ";

int t;

cin >> t;

delete\_record(t); // Delete the record

break;

}

case 5:

return 0; // Exit the program

default:

cout << "\nInvalid choice! Try again.\n"; // Handle invalid choices

}

} while (ch != 5); // Repeat the menu until the user chooses to exit

return 0;

}