Roll No. COA235

Assignment no. 12

INPUT :

#include <cstring> #include <fstream> #include <iostream> using namespace std;

typedef struct student { int rollNo;

char name[50]; char div;

char address[100];

} student;

class studentDatabase {

string fileName = "student\_data.dat";

public:

studentDatabase() {

fstream fileObj(fileName); if (fileObj.fail()) {

fileObj.open(fileName, ios::out); cout << "New file created." << endl;

} else {

cout << "File already exists." << endl;

}

fileObj.close();

}

void addStudent(); void searchStudent(); void deleteStudent(); void displayAll();

};

void studentDatabase::searchStudent() { int roll;

student s;

bool status = false;

// take input of roll number to delete cout << "Roll number to search:\t"; cin >> roll;

// opening files to delete a record ifstream readFile;

readFile.open(fileName, ios::in | ios::binary);

// looking for record

while (readFile.read((char\*)&s, sizeof(student))) { if (s.rollNo == roll) {

status = true; break;

}

}

readFile.close(); if (status) {

cout << "--- RECORD FOUND ---" << endl;

cout << "Roll number:\t" << s.rollNo << endl; cout << "Name:\t" << s.name << endl;

cout << "Division:\t" << s.div << endl; cout << "Address:\t" << s.address << endl;

cout << "--- END OF RECORD ---" << endl;

} else {

cout << "Record not found." << endl;

}

}

void studentDatabase::deleteStudent() { int roll;

student s;

bool status = false;

// take input of roll number to delete cout << "Roll number to delete:\t"; cin >> roll;

// opening files to delete a record ifstream readFile;

readFile.open(fileName, ios::in | ios::binary); ofstream writeFile;

writeFile.open("~" + fileName, ios::out | ios::binary); writeFile.clear();

// looking for record

while (readFile.read((char\*)&s, sizeof(student))) { if (s.rollNo == roll) {

status = true;

} else {

writeFile.write((char\*)&s, sizeof(student)) << flush;

}

}

readFile.close(); writeFile.close();

// moving temp file back to original file if (status) {

readFile.open("~" + fileName, ios::in | ios::binary); writeFile.open(fileName, ios::out | ios::binary); writeFile.clear();

writeFile << readFile.rdbuf(); readFile.close();

writeFile.close();

// remove("~"+fileName);

cout << "Record deleted." << endl;

} else {

cout << "Record not found." << endl;

}

}

void studentDatabase::addStudent() { student s;

cout << "Roll number:\t"; cin >> s.rollNo;

cout << "Name:\t"; cin.ignore(); cin.getline(s.name, 50); cout << "Division:\t";

// cin.ignore(); cin >> s.div;

cout << "Address:\t"; cin.ignore(); cin.getline(s.address, 100);

// cin.ignore();

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

// file.seekp(ios::end);

file.write((char\*)&s, sizeof(student)) << flush; if (file.fail()) {

cout << "Failed to add student record." << endl;

} else {

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

}

file.close();

}

void studentDatabase::displayAll() { ifstream file;

student s;

int count = 0;

file.open(fileName, ios::in | ios::binary); while (file.read((char\*)&s, sizeof(student))) {

count++;

cout << count << ") "; cout << s.rollNo << "|"; cout << s.name << "|"; cout << s.div << "|";

cout << s.address << endl;

}

if (count == 0) {

cout << "No records found." << endl;

}

file.close();

}

int main() { int ch;

studentDatabase db;

// loop do {

cout << endl;

cout << "--- MAIN MENU ---" << endl;

cout << "1 -> Add record" << endl; cout << "2 -> Delete record" << endl; cout << "3 -> Search record" << endl;

cout << "4 -> Display all records" << endl; cout << "0 -> Exit" << endl << endl;

cout << "Choose an option (0-4):\t"; cin >> ch;

switch (ch) { case 0:

cout << "\n\n// END OF CODE\n\n" << endl; break;

case 1:

db.addStudent(); break;

case 2:

db.deleteStudent(); break;

case 3:

db.searchStudent(); break;

case 4:

cout << "All records are:\t" << endl; db.displayAll();

break;

default:

cout << "Please choose a valid option (0-4):\t" << endl; break;

}

} while (ch != 0);

return 0;

}

// END OF CODE

OUTPUT :



