

Assignment No. 10

- **TITLE :** Implement a program for maintaining a database of student records using Files. Students have Student_id,name, Roll_no, Class, marks and address. Display the data for a few students. 1. Create Database 2. Display Database 3. Delete Records 4. Update Record 5. Search Record

➤ **CODE:**

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

class Student {
public:
    string name;
    int roll;
    int marks;
    string division;
    void displayInfo() {
        cout<<name<<" "<<roll<<" "<<marks<<" "<<division<<endl;
    }
    void getInfo() {
        cout << "Enter Name: ";
        cin >> name;
        cout << "Enter Roll Number: ";
        cin >> roll;
        cout << "Enter Marks: ";
        cin >> marks;
        cout << "Enter Division: ";
        cin >> division;
    }
    void insertNewRecord(){
        fstream fout("students.txt", ios::app);
        fout << name << " " << roll << " " << marks << " " << division <<
endl;
        fout.close();
    }
    void displayAllRecords(){
        fstream fin("students.txt", ios::in);
        string n, d;
        int r, m;
        cout << "\n--- Student Records ---" << endl;
        while(fin >> n >> r >> m >> d){
            cout << n << " " << r << " " << m << " " << d << endl;
        }
        fin.close();
    }
    void searchRecord(int sroll){
        cout << "\nSearching for Roll Number: " << sroll << endl;
        fstream fin("students.txt", ios::in);
        string n, d;
        int r, m;
        while(fin >> n >> r >> m >> d){
```

```

        if(r==sroll){
            cout << "Record Found: " << n << " " << r << " " << m << " "
<< d << endl;
            fin.close();
            return;
        }else{
            cout << "Record Not Found" << endl;
        }
    }
}

void deleteRecord(int sroll){
    fstream fin("students.txt", ios::in);
    fstream fout("temp.txt", ios::out);
    string n, d;
    int r, m;
    while(fin >> n >> r >> m >> d){
        if(r!=sroll){
            fout << n << " " << r << " " << m << " " << d << endl;
        }
    }

    fin.close();
    fout.close();
    remove("students.txt");
    rename("temp.txt", "students.txt");
    cout << "Record with Roll Number " << sroll << " deleted." << endl;
}

void updateRecord(int sroll){
    fstream fin("students.txt", ios::in);
    fstream fout("temp.txt", ios::out);
    string n, d;
    int r, m;
    while(fin >> n >> r >> m >> d){
        if(r==sroll){
            cout << "Enter New Details for Roll Number " << sroll <<
":\n";

            getInfo();
            fout << name << " " << roll << " " << marks << " " <<
division << endl;
        }else{
            fout << n << " " << r << " " << m << " " << d << endl;
        }
    }
    fin.close();
    fout.close();
    remove("students.txt");
    rename("temp.txt", "students.txt");
    cout << "Record with Roll Number " << sroll << " updated." << endl;
}

};

int main() {
    Student student;
    int ch, rollNo;
    while(true){

```

```

        cout << "\n1. Insert New Record\n2. Display All Records\n3. Search
Record\n4. Delete Record\n5. Update Record\n6. Exit\nEnter your choice: ";
        cin >> ch;
        if(ch==6) break;
        switch(ch){
            case 1:
                student.getInfo();
                student.insertNewRecord();
                break;
            case 2:
                student.displayAllRecords();
                break;
            case 3:
                cout << "Enter Roll Number to Search: ";
                cin >> rollNo;
                student.searchRecord(rollNo);
                break;
            case 4:
                cout << "Enter Roll Number to Delete: ";
                cin >> rollNo;
                student.deleteRecord(rollNo);
                break;
            case 5:
                cout << "Enter Roll Number to Update: ";
                cin >> rollNo;
                student.updateRecord(rollNo);
                break;
            default:
                cout << "Invalid Choice!" << endl;
        }
    }
}

```

Output –

```

1. Insert New Record
2. Display All Records
3. Search Record
4. Delete Record
5. Update Record
6. Exit
Enter your choice: 1
Enter Name: Nisha
Enter Roll Number: 13
Enter Marks: 89
Enter Division: SEDA

```

```

1. Insert New Record
2. Display All Records
3. Search Record
4. Delete Record
5. Update Record
6. Exit
Enter your choice: 2
--- Student Records ---
Nisha 13 89 SEDA

```

```

students.txt X
students.txt
1  Nisha 13 89 SEDA
2

```

```

1. Insert New Record
2. Display All Records
3. Search Record
4. Delete Record
5. Update Record
6. Exit
Enter your choice: 3
Enter Roll Number to Search: 13

Searching for Roll Number: 13
Record Found: Nisha 13 89 SEDA

```

```

1. Insert New Record
2. Display All Records
3. Search Record
4. Delete Record
5. Update Record
6. Exit
Enter your choice: 4
Enter Roll Number to Delete: 24
Record with Roll Number 24 deleted.

```

```

students.txt X
students.txt
1  Nisha 13 89 SEDA
2

```

```

1. Insert New Record
2. Display All Records
3. Search Record
4. Delete Record
5. Update Record
6. Exit
Enter your choice: 5
Enter Roll Number to Update: 24
Enter New Details for Roll Number 24:
Enter Name: Sohaa
Enter Roll Number: 24
Enter Marks: 95
Enter Division: SEDA
Record with Roll Number 24 updated.

```

```

students.txt
1  Nisha 13 89 SEDA
2  Soha 24 90 SEDA
3

```

```

students.txt
1  Nisha 13 89 SEDA
2  Sohaa 24 95 SEDA
3

```

- **Conclusion** – In this assignment, we've implemented program on implementation Of handling student data using file handling operations and methods.