Code :

```
#include <bits/stdc++.h>
#include <iomanip>
using namespace std;

class Student
{
private:
    double RollNo;
    string name;
    string division;
    string address;

public:
    Student() {}

    friend ofstream &operator<<(ofstream &ofs, Student &s);
    friend ifstream &operator>>(ifstream &ifs, Student &s);
    friend ostream &operator<<(ostream &os, Student &s);
    friend istream &operator>>(istream &is, Student &s);
    static void Create();
    static void Display();
    static void Add();
    static void Delete();
    static void Update();
    static void Search();
    static int getNoofLine(string filename);
};

void Student ::Create()
{
    ofstream outfile("student.txt", ios::trunc);
    cout << "The database is created !" << endl;
}

void Student ::Display()
{

    ifstream infile("student.txt", ios::in);
    infile.clear();
    infile.seekg(0);
    int n = getNoofLine("student.txt");
    cout << "Roll No   "
         << "Name   "
         << "Division   "
         << "Address" << endl;
    for (int i = 0; i < n; i++)
    {
        Student s;
        infile >> s;
        cout << s << "\n";
    }

    infile.close();
}

void Student ::Add()
{
    ofstream outfile("student.txt", ios::app);
    cout << "Enter no of entries that you want to add : ";
    int n;
    cin >> n;
    Student s[n];
    for (int i = 0; i < n; i++)
    {
        cin >> s[i];
        outfile << s[i];
    }
    cout << endl;
    outfile.close();
}

void Student ::Delete()
{
    ifstream infile("student.txt", ios::in);
    ofstream outfile("new.txt", ios::out);
    cout << "Enter the Roll No that you want to delete : ";
    double key;
    cin >> key;
    bool flag = true;
    int n = getNoofLine("student.txt");
    for (int i = 0; i < n; i++)
    {
        Student s;
        infile >> s;

        if (s.RollNo == key)
        {
            flag = false;
            continue;
        }
        else
        {
            outfile << s;
        }
    }
    if (flag)
    {
        cout << "Student Not Preset in Database\n";
    }
    else
    {
        infile.close();
        outfile.close();
        remove("student.txt");
        rename("new.txt", "student.txt");

        cout << "Record deleted\n";
    }
}

void Student ::Search()
{
    cout << "Enter the Roll No that you want to Search : ";
    double key;
    cin >> key;
    bool flag = true;
    ifstream infile("student.txt");
    int n = getNoofLine("student.txt");
    for (int i = 0; i < n; i++)
    {
        Student s;
        infile >> s;
        if (s.RollNo == key)
        {
            flag = false;
            cout << s << "\n";
            return;
        }
    }
    if (flag)
    {
        cout << "Student not found\n";
    }
}

void Student::Update()
{
    ifstream infile("student.txt");
    ofstream outfile("new.txt");
    bool flag = true;

    cout << "Enter the details to be updated" << endl;
    Student s2;
    cin >> s2;
    int n = getNoofLine("student.txt");

    for (int i = 0; i < n; i++)
    {
        Student s;
        infile >> s;
        if (s.RollNo == s2.RollNo)
        {
            flag = false;
            cout << "Updating the data\n";
            outfile << s2;
        }
        else
        {
            outfile << s;
        }
    }
    if (flag)
    {
        cout << "Student not found\n";
    }
    else
    {
        infile.close();
        outfile.close();
        remove("student.txt");
        rename("new.txt", "student.txt");
    }
}

int Student::getNoofLine(string filename)
{
    ifstream infile(filename, ios::in);
    int count = 0;
    string line;
    if (infile.is_open())
    {
        while (getline(infile, line))
        {
            count++;
        }
    }
    return count;
}

ofstream &operator<<(ofstream &ofs, Student &s)
{
    ofs << s.RollNo << " " << s.name << " " << s.division << " " << s.address <<
endl;
    return ofs;
}

ifstream &operator>>(ifstream &ifs, Student &s)
{
    ifs >> s.RollNo >> s.name >> s.division >> s.address;
    return ifs;
}

ostream &operator<<(ostream &os, Student &s)
{
    os << s.RollNo << " " << s.name << " " << s.division << " " << s.address;
    return os;
}

istream &operator>>(istream &is, Student &s)
{
    cout << "Enter Your Roll No : ";
    is >> s.RollNo;
    cin.ignore(1); // dont take string input directly after int input
    cout << "Enter Your Name : ";
    getline(is, s.name);
    cout << "Enter Your Division : ";
    getline(is, s.division);
    cout << "Enter Your Address : ";
    getline(is, s.address);
    return is;
}

int main()
{
    cout << "Please select a choice :";
    int ch;
    do
    {
        cout << endl;
        cout << "Choose what operation you want\n";
        cout << "1) Create a new students database\n";
        cout << "2) Display students database\n";
        cout << "3) Insert new entries\n";
        cout << "4) Update entries\n";
        cout << "5) Delete entry\n";
        cout << "6) Search an entry\n";
        cout << "7) Exit\n";
        cout << "Please select a choice :";
        cin >> ch;
        cout << endl;
        int n;
        if (ch == 1)
        {
            Student::Create();
        }
        else if (ch == 2)
        {
            Student::Display();
        }
        else if (ch == 3)
        {
            Student::Add();
        }
        else if (ch == 4)
        {
            Student::Update();
        }
        else if (ch == 5)
        {
            Student::Delete();
        }
        else if (ch == 6)
        {
            Student::Search();
        }
        else if (ch == 7)
        {
            cout << "Program finished\n";
            break;
        }
        else
        {
            cout << "Enter correct option\n";
        }

    } while (ch != 7);

    return 0;
}
```

Output :

```
Please select a choice :
Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :1

The database is created !
```

```
Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :3

Enter no of entries that you want to add : 10
Enter Your Roll No : 67
Enter Your Name : Parth
Enter Your Division : 9
Enter Your Address : Amalner
Enter Your Roll No : 58
Enter Your Name : Aditya
Enter Your Division : 4
Enter Your Address : Jalgaon
Enter Your Roll No : 1
Enter Your Name : Shivraj
Enter Your Division : 3
Enter Your Address : Nashik
Enter Your Roll No : 14
Enter Your Name : Vedant
Enter Your Division : 5
Enter Your Address : Pune
Enter Your Roll No : 56
Enter Your Name : Tushar
Enter Your Division : 1
Enter Your Address : Kalyan
Enter Your Roll No : 10
Enter Your Name : Harsh
Enter Your Division : 11
Enter Your Address : Surat
Enter Your Roll No : 16
Enter Your Name : Krishna
Enter Your Division : 2
Enter Your Address : Andheri
Enter Your Roll No : 20
Enter Your Name : Halide
Enter Your Division : 9
Enter Your Address : Nagpur
Enter Your Roll No : 52
Enter Your Name : Hitesh
Enter Your Division : 10
Enter Your Address : Nashik
Enter Your Roll No : 51
Enter Your Name : Ayush
Enter Your Division : 3
Enter Your Address : Satara


Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
```

```
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :4

Enter the details to be updated
Enter Your Roll No : 52
Enter Your Name : Hitesh
Enter Your Division : 5
Enter Your Address : Nashik
Updating the data

Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :2

Roll No   Name   Division   Address
67 Parth 9 Amalner
58 Aditya 4 Jalgaon
1 Shivraj 3 Nashik
14 Vedant 5 Pune
56 Tushar 1 Kalyan
10 Harsh 11 Surat
16 Krishna 2 Andheri
20 Halide 9 Nagpur
52 Hitesh 5 Nashik
51 Ayush 3 Satara

Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :4

Enter the details to be updated
Enter Your Roll No : 51
Enter Your Name : Ayush
Enter Your Division : 4
Enter Your Address : Nagpur
Updating the data

Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
```

```
Please select a choice :2

Roll No   Name   Division   Address
67 Parth 9 Amalner
58 Aditya 4 Jalgaon
1 Shivraj 3 Nashik
14 Vedant 5 Pune
56 Tushar 1 Kalyan
10 Harsh 11 Surat
16 Krishna 2 Andheri
20 Halide 9 Nagpur
52 Hitesh 5 Nashik
51 Ayush 4 Nagpur

Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :5

Enter the Roll No that you want to delete : 20
Record deleted

Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :2

Roll No   Name   Division   Address
67 Parth 9 Amalner
58 Aditya 4 Jalgaon
1 Shivraj 3 Nashik
14 Vedant 5 Pune
56 Tushar 1 Kalyan
10 Harsh 11 Surat
16 Krishna 2 Andheri
52 Hitesh 5 Nashik
51 Ayush 4 Nagpur

Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :6

Enter the Roll No that you want to Search : 67
```

```
67 Parth 9 Amalner

Choose what operation you want
1) Create a new students database
2) Display students database
3) Insert new entries
4) Update entries
5) Delete entry
6) Search an entry
7) Exit
Please select a choice :7

Program finished
```