



Bansilal Ramnath Agarwal Charitable Trust's
Vishwakarma Institute of Information
Technology

**Department of
Artificial Intelligence and Data
Science**

Name: Siddhesh Dilip Khairnar

Class: SY

Division: B

Roll No: 272028

Semester: IV

Academic Year: 2022-2023

Subject Name & Code: Advance Data Structure: ADUA22202

Title of Assignment: Implement student database by implementing sequential file organization.

Date of Performance: 26-04-2023

Date of Submission: 26-04-2023

ASSIGNMENT NO. 7

Program and Output:

```
#include <iostream>
#include <fstream>
#include <string.h>
using namespace std;
class Student
{
    char Name[10];
    int Roll_No;

public:
    Student()
    {
        Name[0] = '\0';
        Roll_No = -1;
    }
    void get_Data();
    void put_Data();
    int return_Roll() { return Roll_No; }
};

void Student::get_Data()
{
    cout << "\nEnter Student Data";
    cout << "\nName";
    cin >> Name;
    cout << "\nRoll Number: ";
    cin >> Roll_No;
}

void Student::put_Data()
{
    cout << "\n"
         << Roll_No << "\t" << Name;
}

class Seq_File
{
    char File_Name[15];

public:
    Seq_File();
    Seq_File(char F[]);
    void Create();
    void Display();
    void Add();
    void Remove(int);
    void Modify(int);
    void Search(int);
};
```

```

};
Seq_File::Seq_File()
{
    ofstream File("Student.txt");
    strcpy(File_Name, " Student.txt");
    cout << "\nDefault Constructor";
    if (File)
    {
        cout << "\nFile opened Successfully";
        File.close();
    }
    else
        cout << "\nFile creation Error";
}
Seq_File::Seq_File(char F[15])
{
    ofstream File;
    strcpy(File_Name, F);
    File.open(F);
    if (File)
    {
        cout << "\nFile opened Successfully";
        File.close();
    }
    else
        cout << "\nFile creation Error";
}
void Seq_File::Create()
{
    ofstream File;
    Student S;
    File.open(File_Name);
    S.get_Data();
    File.write(reinterpret_cast<char *>(&S), sizeof(S));
    File.close();
}
void Seq_File::Display()
{
    ifstream File;
    Student S;
    File.open(File_Name);
    cout << "\nRoll No\t Student Name";
    File.read(reinterpret_cast<char *>(&S), sizeof(S));
    while (!File.eof())
    {
        S.put_Data();
        File.read(reinterpret_cast<char *>(&S), sizeof(S));
    }
}

```

```

    }
    File.close();
}
void Seq_File::Add()
{
    ofstream File;
    Student S;
    File.open(File_Name, ios::app);
    S.get_Data();
    File.write(reinterpret_cast<char *>(&S), sizeof(S));
    File.close();
}
void Seq_File::Remove(int Roll)
{
    ifstream File;
    ofstream Temp;
    Student S;
    int Flag = 0;
    File.open(File_Name);
    Temp.open("Temp.Txt");
    File.read(reinterpret_cast<char *>(&S), sizeof(S));
    while (!File.eof())
    {
        if (Roll == S.return_Roll())
        {
            S.put_Data();
            Flag = 1;
        }
        else
            Temp.write(reinterpret_cast<char *>(&S), sizeof(S));
        File.read(reinterpret_cast<char *>(&S), sizeof(S));
    }
    if (Flag == 0)
        cout << "Roll No. " << Roll << " does not present \n";
    File.close();
    Temp.close();
    remove(File_Name);
    rename("Temp.Txt", File_Name);
}
void Seq_File::Modify(int Roll)
{
    ifstream File;
    ofstream Temp;
    Student S;
    int Flag = 0;
    File.open(File_Name);
    Temp.open("Temp.Txt");
    File.read(reinterpret_cast<char *>(&S), sizeof(S));

```

```

while (!File.eof())
{
    if (Roll == S.return_Roll())
    {
        S.put_Data();
        cout << "\n Enter data to modify";
        S.get_Data();
        Flag = 1;
    }
    Temp.write(reinterpret_cast<char *>(&S), sizeof(S));
    File.read(reinterpret_cast<char *>(&S), sizeof(S));
}
if (Flag == 0)
    cout << "Roll No. " << Roll << " does not present \n";
File.close();
Temp.close();
remove(File_Name);
rename("Temp.Txt", File_Name);
}

void Seq_File::Search(int Roll)
{
    ifstream File;
    Student S;
    File.open(File_Name);
    int flag = 0;
    File.read(reinterpret_cast<char *>(&S), sizeof(S));
    while (!File.eof())
    {
        if (Roll == S.return_Roll())
        {
            S.put_Data();
            flag = 1;
        }

        File.read(reinterpret_cast<char *>(&S), sizeof(S));
    }

    if (flag == 0)
    {
        cout << "\n Record Doesn't Exist";
    }

    File.close();
}

int main()
{

```

```

int Choice;
char F[15];
int R;
cout << "\nEnter File Name : ";
cin >> F;
Seq_File sFile(F);
do
{
    cout << "\n1: Create Database\n2: Display Database\n3: Add a
record\n4: Delete a record\n5: Modify a record\n 6:Search a Record \n Enter
your choice: ";
    cin >> Choice;
    switch (Choice)
    {
        case 1:
            sFile.Create();
            break;
        case 2:
            sFile.Display();
            break;
        case 3:
            sFile.Add();
            break;
        case 4:
            cout << "\nEnter Roll No to delete";
            cin >> R;
            sFile.Remove(R);
            break;
        case 5:
            cout << "\nEnter Roll No to Modify";
            cin >> R;
            sFile.Modify(R);
            break;

        case 6:
            cout << "\n Enter Roll no to search record";
            cin >> R;
            sFile.Search(R);
            break;
    }
} while (Choice < 7);
cout << "\n";
return 1;
}

```

Enter File Name : B

File opened Successfully

1: Create Database

2: Display Database

3: Add a record

4: Delete a record

5: Modify a record

6:Search a Record

Enter your choice: 1

Enter Student Data

NameSiddhesh

Roll Number: 28

1: Create Database

2: Display Database

3: Add a record

4: Delete a record

5: Modify a record

6:Search a Record

Enter your choice: 2

Roll No Student Name

28 Siddhesh

1: Create Database

2: Display Database

3: Add a record

4: Delete a record

5: Modify a record

6:Search a Record

Enter your choice: 2

Roll No Student Name

28 Siddhesh

1: Create Database

2: Display Database

3: Add a record

4: Delete a record

5: Modify a record

6:Search a Record

Enter your choice: 4

Enter Roll No to delete28

Roll No Student Name

21 om

1: Create Database

2: Display Database

3: Add a record

4: Delete a record

5: Modify a record

6:Search a Record

Enter your choice: