Assignment 10

Title: File Handling.

Problem Statement: Department maintains student information. The file contains roll number, name, division and address. Allow user to add, delete, search student data. If record does not exist print particular message. Use sequential file for implementing file handling.

Objective: To study and implement file handling operations.

Outcome: Study and implement file handling operations.

Theory:
File is a stream of bytes. Size of file is expressed in number of bytes.
Sequential file:
In this file, records are stored in order of arrival Length of record is not fixed search is time consuming. Insertion and deletion of record is also time consuming.
Random files:
We can read/write a particular word without having read previous records. We can position file pointer to a specific location.
Modes:

- ios::in (Read from file)
- ios::out (Write to file)
- ios::ate (go to end of file)
- ios::app (append to end of file)
- ios::trunk (truncate file)
- ios::binary (treat file as binary)

Class Definition:

```
class Student
{
        int roll_no;
        string name;
        string temp, address;
        char division;
public:
        void create();
        void insert();
        void display();
        void delet();
        void find();
};
```

Pseudo Codes:

```
void Student::create()
{
        int n;
        cout << "Enter number of entries: ";
        cin >> n;
        ofstream fout;
        fout.open("Student.txt", ios::out|ios::app);
        for(int i=0;i<n;i++)
        {
                cout << "Enter roll number: ";
                cin >> roll_no;
                cin.ignore(1);
                cout << "Enter name: ";
                getline(cin,name);
                cout << "Enter division: ";
                cin >> division;
                cin.ignore(1);
                cout << "Enter address: ";
                getline(cin,address);
                fout << name << "\t" << roll_no << "\t" << division << "\t" <<
address << "\n";
        }
        fout.close();
}
```

```
void Student::insert()
{
	cout << "Enter roll number: ";
	cin >> roll_no;
	cin.ignore(1);
	cout << "Enter name: ";
	getline(cin,name);
	cout << "Enter division: ";
	cin >> division;
	cin.ignore(1);
	cout << "Enter address: ";
	getline(cin,address);
	ofstream fout;
	fout.open("Student.txt", ios::out|ios::app);
	fout << name << "\t" << roll_no << "\t" << division << "\t" << address <<
"\n";
	fout.close();
}

void Student::display()
{
	ifstream fin;
	fin.open("Student.txt",ios::in|ios::app);
	cout << "Name\tRollNo\tDiv\tAddress";
	string temp;
	fin.seekg(0,ios::beg);
	while(fin.eof() == 0)
	{
		getline(fin,temp);
		cout << "\n" << temp;
	}
	fin.close();
}

void Student::delet()
{
	ifstream fin;
	fin.open("Student.txt",ios::in|ios::app);
	string temp,name,temp2,temp3;
```

```
   cout << "\nEnter name of student whose record is to be deleted: ";
   getline(cin,name);
   fin.seekg(0,ios::beg);
   int flag = 0;
   ofstream fout;
   fout.open("yolo.txt", ios::out);
   int cnt = 0;
   while(fin.eof() == 0 && fin.tellg() != -1)
   {
            fin >> temp;
            if(temp == name)
            {
                     flag = 1;
            }
            if(flag == 1)
            {
                     for(int i=0;i<4;i++)
                     {
                              fin >> temp;
                     }
                     flag = 0;
   cnt = 0;
            }
            fout << temp;
            if(cnt == 3)
{
   cnt = 0;
   fout << "\n";
}
else
{
   cnt++;
   fout << "\t";
}
   }
   fout.close();
   fin.close();
   cnt = 0;
   fin.open("yolo.txt",ios::in|ios::app);
   fout.open("Student.txt",ios::out | ios::trunc);
```

```
        while(fin.eof() == 0)
        {
                fin >> temp2;
                if(temp2 != temp3)
                {
                        fout << temp2;
                        if(cnt == 3)
                        {
                           cnt = 0;
                           fout << "\n";
                        }
                        else
                        {
                           cnt++;
                           fout << "\t";
                        }
                }
                temp3 = temp2;
        }
        fin.close();
        fout.close();
}

void Student::find()
{
   ifstream fin;
        fin.open("Student.txt",ios::in|ios::app);
        string temp,name;
        cout << "\nEnter name of student whose record is to be searched: ";
        getline(cin,name);
        fin.seekg(0,ios::beg);
        int flag = 0;
        while(!fin.eof())
        {
                fin >> temp;
                if(temp == name)
                {
                        flag = 1;
                        break;
                }
```

```
        }
        if(flag == 1)
        {
      for(int i=0;i<4;i++)
                {
                   cout << temp << "\t";
                        fin >> temp;
                }
                fin.close();
                return;
        }
        cout << "Record not found.\n";
}
```

Testcases:

1. Input:

Roll no: 1, 22, 33
Name: a, b, c
Div: 1, 5, 7
Expected O/P:
1 a 1
22 b 5
33 c 7
Actual O/P: 1 a 1
22 b 5
33 c 7
Result:
Pass

2. Input:

Search: 22
Expected O/P:
22 b 5
Actual O/P:
22 b 5
Result:
Pass

3. Input:

Delete: 22
Expected O/P:
1 a 1
33 c 7
Actual O/P:
1 a 1
33 c 7
Result:
Pass

Conclusion:
We have successfully implemented file handling operations using sequential file.