**Ganpat University**

**Faculty of Engineering & Technology**

**Computer Science & Engineering**

***Name:- Dwij Vatsal Desai***

***Sem:- 2***

***Sub: - ESFP-II***

***Enrollment No.:- 23162121027***

***Prac:- 14***

**Practical 14**

***Definition:***

***Complete the code for the object assigned to you to satisfy the following*** specifications.

1. For the solving purpose of given topic practical, you need to create minimum one class, rest as per your requirement.
2. Declare minimum seven function like addInfo (), displayInfo (), editInfo (), searchInfo (), deleteInfo (), showDataInascOrder (), showDataIndscOrder () and exit () in class.
3. You can you file handling pre-defined class and their related function and mode.
4. Store information in binary format and perform accordingly.
5. Minimum 1 constructor method should be available in the program, rest as per your requirement.
6. You must use access specifier for data member and member function declaration in program.
7. Show all record in ascending and descending order by their id or name.
8. Take minimum 5 data record from the user and display according to the choice of user category wise. (Minimum seven different options should be there for displaying information, and if you want to add more choice option as per your program requirement, you can add it.)

## .Code:-

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

class Student
{
public:
    string Name;
    int enroll_no;
    string course;
    int sem;

public:
    void input()
    {
        cout << "Enter name: ";
        cin >> Name;
        cout << "Enter enrollment number: ";
        cin >> enroll_no;
        cout << "Enter course: ";
        cin >> course;
        cout << "Enter semester: ";
        cin >> sem;
    }

    void display()
    {
        cout << "Name: " << Name << endl;
        cout << "Enrollment Number: " << enroll_no << endl;
        cout << "Course: " << course << endl;
        cout << "Semester: " << sem << endl;
    }

    string getName()
    {
        return Name;
    }

    int getEnrollmentNumber()
    {
        return enroll_no;
    }

    string getCourse()
    {
        return course;
    }

    int getSemester()
    {
        return sem;
    }
};

void addData()
{
    Student stu;
    stu.input();

    ofstream file("data.bin", ios::app | ios::binary);
    if (file.is_open())
    {
```

```
        file << stu.getName() << " " << stu.getEnrollmentNumber() << " " <<
stu.getCourse() << " " << stu.getSemester() << endl;
        file.close();
        cout << "Data added successfully!" << endl;
    }
    else
    {
        cout << "Unable to open file." << endl;
    }
}

void viewData()
{
    ifstream file("data.bin", ios::binary);
    if (file.is_open())
    {
        string name;
        int enroll_no;
        string course;
        int sem;
        while (file >> name >> enroll_no >> course >> sem)
        {
            cout << "Name: " << name << ", Enrollment Number: " << enroll_no << ",
Course: " << course << ", Semester: " << sem << endl;
        }
        file.close();
    }
    else
    {
        cout << "Unable to open file." << endl;
    }
}

void deleteData()
{
    string name;
    cout << "Enter name to delete: ";
    cin >> name;

    ifstream file("data.bin");
    if (file.is_open())
    {
        ofstream temp("temp.bin");
        string n;
        int enroll_no;
        string course;
        int sem;
        while (file >> n >> enroll_no >> course >> sem)
        {
            if (n != name)
            {
                temp << n << " " << enroll_no << " " << course << " " << sem <<
endl;
            }
        }
        file.close();
        temp.close();
        remove("data.bin");
        rename("temp.bin", "data.bin");
        cout << "Data deleted successfully!" << endl;
    }
    else
    {
        cout << "Unable to open file." << endl;
    }
}

void updateData()
```

```
{
    string name;
    cout << "Enter name to update: ";
    cin >> name;

    ifstream file("data.bin");
    if (file.is_open())
    {
        ofstream temp("temp.bin");
        string n;
        int enroll_no;
        string course;
        int sem;
        while (file >> n >> enroll_no >> course >> sem)
        {
            if (n != name)
            {
                temp << n << " " << enroll_no << " " << course << " " << sem <<
endl;
            }
            else
            {
                Student stu;
                stu.input();
                temp << stu.getName() << " " << stu.getEnrollmentNumber() << " " <<
stu.getCourse() << " " << stu.getSemester() << endl;
            }
        }
        file.close();
        temp.close();
        remove("data.bin");
        rename("temp.bin", "data.bin");
        cout << "Data updated successfully!" << endl;
    }
    else
    {
        cout << "Unable to open file." << endl;
    }
}

void searchData()
{
    string name;
    cout << "Enter name to search: ";
    cin >> name;

    ifstream file("data.bin");
    if (file.is_open())
    {
        string n;
        int enroll_no;
        string course;
        int sem;
        bool found = false;
        while (file >> n >> enroll_no >> course >> sem)
        {
            if (n == name)
            {
                cout << "Name: " << n << ", Enrollment Number: " << enroll_no << ",
Course: " << course << ", Semester: " << sem << endl;
                found = true;
                break;
            }
        }
        file.close();
        if (!found)
        {
            cout << "Data not found." << endl;
```

```
        }
    }
    else
    {
        cout << "Unable to open file." << endl;
    }
}

int main() {
    int choice;
    do
    {
        cout << "1. Add data" << endl;
        cout << "2. View data" << endl;
        cout << "3. Delete data" << endl;
        cout << "4. Update data" << endl;
        cout << "5. Search data" << endl;
        cout << "6. Sort data" << endl;
        cout << "0. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> choice;

        Student stu[10]; // Move the declaration and initialization here

        switch (choice)
        {
            case 1:
                addData();
            break;
            case 2:
                viewData();
            break;
            case 3:
                deleteData();
            break;
            case 4:
                updateData();
            break;
            case 5:
                searchData();
            break;
            case 6:
            {
                ifstream file("data.bin");
                if (file.is_open())
                {
                    Student stu[10];
                    int count = 0;

                    // Read data into array
                    while (count < 10 && file >> stu[count].Name >>
stu[count].enroll_no >> stu[count].course >> stu[count].sem)
                    {
                        count++;
                    }
                    file.close();

                    // Sort the array based on name
                    int input;
                    cout << "1. Ascending." << endl;
                    cout << "2. Descending." << endl;
                    cout << "Enter choice: ";
                    cin >> input;

                    Student temp;
                    for (int i = 0; i < count; i++)
                    {
                        for (int j = i + 1; j < count; j++)
```

```
                        {
                            string comp = stu[i].Name;
                            string comp1 = stu[j].Name;
                            if ((comp.compare(comp1) > 0 && input == 1) ||
(comp.compare(comp1) < 0 && input == 2))
                            {
                                temp = stu[i];
                                stu[i] = stu[j];
                                stu[j] = temp;
                            }
                        }
                    }

                    // Display the sorted array
                    for (int i = 0; i < count; i++)
                    {
                        cout << "Name: " << stu[i].Name << ", Enrollment Number: "
<< stu[i].enroll_no << ", Course: " << stu[i].course << ", Semester: " <<
stu[i].sem << endl;
                    }
                }
                else
                {
                    cout << "Unable to open file." << endl;
                }
                break;
            }

            case 0:
                cout <<"Exiting..." << endl;
            break;
            default:
                cout << "Invalid choice. Please try again." << endl;
            break;
        }
    } while (choice != 0);
    return 0;
}
```

## ***Output:-***

```
"C:\Users\dwijd\OneDrive\Documents\collage practicals\ESFP-II\Practical_14.exe"
1. Add data
2. View data
3. Delete data
4. Update data
5. Search data
6. Sort data
0. Exit
Enter your choice:1
 Enter name:Dwij
 Enter enrollment number:27
 Enter course:CS
 Enter semester:2
 Data added successfully!
1. Add data
2. View data
3. Delete data
4. Update data
5. Search data
6. Sort data
0. Exit
Enter your choice:1
 Enter name:Ender
 Enter enrollment number:28
 Enter course:BTech
 Enter semester:3
 Data added successfully!
1. Add data
2. View data
3. Delete data
4. Update data
5. Search data
6. Sort data
0. Exit
Enter your choice:2
 Name: Dwij, Enrollment Number: 27, Course: CS, Semester: 2
Name: Ender, Enrollment Number: 28, Course: BTech, Semester: 3
1. Add data
2. View data
3. Delete data
4. Update data
5. Search data
6. Sort data
0. Exit
Enter your choice:4
 Enter name to update:Ender
 Enter name:Emder
 Enter enrollment number:16
 Enter course:MTech
 Enter semester:4
 Data updated
 successfully!
```

```
1. Add data
2. View data
3. Delete data
4. Update data
5. Search data
6. Sort data
0. Exit
Enter your choice:6
 1. Ascending.
2. Descending.
Enter choice:1
 Name: Dwij, Enrollment Number: 27, Course: CS, Semester: 2
Name: Emder, Enrollment Number: 16, Course: MTech, Semester: 4
1. Add data
2. View data
3. Delete data
4. Update data
5. Search data
6. Sort data
0. Exit
Enter your choice:6
 1. Ascending.
2. Descending.
Enter choice:2
 Name: Emder, Enrollment Number: 16, Course: MTech, Semester: 4
Name: Dwij, Enrollment Number: 27, Course: CS, Semester: 2
1. Add data
2. View data
3. Delete data
4. Update data
5. Search data
6. Sort data
0. Exit
Enter your choice:0
 Exiting...

Process finished with exit code 0
```