```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Policy;
using System.Text;
using System.Threading.Tasks;

namespace OSL_Assignment3
{

    public class Student
    {
        string name, section;
        public string Name { get { return name; } set { name = value; } }
        public string Section { get { return section; } set { section = value; } }
        public Student(string name,string section)
        {
            this.Name = name;
            this.Section = section;
        }
    }
    public class Teacher
    {
        string name, section;
        public string Name { get { return name; } set { name = value; } }
```

```
    public string Section { get { return section; } set { section = value; } }
    public Teacher(string name, string section)
    {
        this.Name = name;
        this.Section = section;
    }
}
public class Subject
{
    //string subName, subCode, teacher;
    public string SubName { get; set; }
    public int SubCode { get; set; }
    public string Teacher { get; set; }
    public Subject(string name,int code,string teacher)
    {
        SubName = name;
        SubCode  = code;
        Teacher = teacher;
    }
}
internal class Program
{
    static List<Student> students = new List<Student>();
    static List<Teacher> teachers = new List<Teacher>();
    static List<Subject> subjects = new List<Subject>();
```

```
public static void AddData()
{
    students.Add(new Student("John", "10th"));
    students.Add(new Student("Jane", "9th"));
    students.Add(new Student("Smitha", "12th"));
    students.Add(new Student("Raj", "11th"));
    students.Add(new Student("Priya", "8th"));
    students.Add(new Student("Kumar", "11th"));
    students.Add(new Student("Sara", "10th"));
    students.Add(new Student("Vikram", "12th"));
    students.Add(new Student("Deepa", "9th"));
    students.Add(new Student("Anand", "8th"));
    students.Add(new Student("Karthik", "12th"));
    students.Add(new Student("Sundari", "10th"));
    students.Add(new Student("Vijay", "9th"));
    students.Add(new Student("Gowri", "11th"));
    students.Add(new Student("Arjun", "8th"));
    students.Add(new Student("Saranya", "11th"));
    students.Add(new Student("Krishna", "10th"));
    students.Add(new Student("Manikandan", "12th"));
    students.Add(new Student("Suresh", "9th"));
    students.Add(new Student("Amutha", "8th"));

    teachers.Add(new Teacher("Mr. Kumar", "10th"));
    teachers.Add(new Teacher("Mrs. Rani", "11th"));
```

```
    teachers.Add(new Teacher("Mr. Balaji", "12th"));
    teachers.Add(new Teacher("Ms. Priya", "9th"));
    teachers.Add(new Teacher("Mr. Rajesh", "8th"));

    subjects.Add(new Subject("Mathematics", 101, "Mr. Kumar"));
    subjects.Add(new Subject("Physics", 102, "Mrs. Rani"));
    subjects.Add(new Subject("Chemistry", 103, "Mr. Balaji"));
    subjects.Add(new Subject("English", 104, "Ms. Priya"));
    subjects.Add(new Subject("Biology", 105, "Mr. Rajesh"));
    subjects.Add(new Subject("Computer Science", 106, "Mr. Kumar"));
    subjects.Add(new Subject("Economics", 107, "Mrs. Rani"));
    subjects.Add(new Subject("History", 108, "Mr. Balaji"));
    subjects.Add(new Subject("Geography", 109, "Ms. Priya"));
    subjects.Add(new Subject("Environmental Science", 110, "Mr. Rajesh"));
}
public static void DisplayStudents()
{
    foreach(var student in students)
    {
        Console.WriteLine($"Student Name : {student.Name}, Class : {student.Section}");
    }
}
public static void DisplayTeachers()
{
    foreach (var teacher in teachers)
```

```
        {
            Console.WriteLine($"Teacher Name : {teacher.Name}, Class : {teacher.Section}");
        }
    }
    public static void DisplaySubjects()
    {
        foreach (var subject in subjects)
        {
            Console.WriteLine($"Subject Name : {subject.SubName}, Subject Code : {subject.SubCode},
Teacher Name : {subject.Teacher}");
        }
    }
    public static void StudentsInClass(string cl)
    {
        int count = 0;
        foreach(var student in students)
        {
            if (student.Section.Equals(cl))
            {
                count++;
                Console.WriteLine($"Student Name : {student.Name}, Class : {student.Section}");
            }
        }
        if (count == 0)
            Console.WriteLine("There is no such class or You mush have entered the wrong class");
    }
```

```
public static void SubjectsTaughtByTeacher(string name)
{
    int count = 0;
    Console.WriteLine($"\nSubject taught by {name}\n");
    foreach (var subject in subjects)
    {
        if (subject.Teacher.Equals(name))
        {
            count++;
            Console.WriteLine($"Subject Name : {subject.SubName}, Subject Code : {subject.SubCode}");
        }
    }
    if (count == 0)
        Console.WriteLine("There is no such teacher or You must have entered the wrong name");
}
static void Main(string[] args)
{
    try
    {
        AddData();
        Console.WriteLine("Choose one option to Perform\n\n1. Display All Data\n2. Display Students
in Specific Class\n3. Display Subject taught by Specific Teacher\n\nEnter the Function number");
        Again:
        switch (int.Parse(Console.ReadLine()))
        {
            case 1:
```

```
    {
        Console.WriteLine("The List of Students\n");
        DisplayStudents();
        Console.WriteLine("\nThe List of Teachers\n");
        DisplayTeachers();
        Console.WriteLine("\nThe List of Subjects\n");
        DisplaySubjects();
        break;
    }
case 2:
    {
        Console.WriteLine("\nEnter the Class");
        string input = Console.ReadLine();
        Console.WriteLine($"\nThe List of Students in Class {input}\n");
        StudentsInClass(input);
        break;
    }
case 3:
    {
        Console.WriteLine("Enter Teacher Name");
        string input = Console.ReadLine();
        SubjectsTaughtByTeacher(input);
        break;
    }
default:
```

```
                {
                    Console.WriteLine("Oops!!!\nWrong Option\nSelect Again");
                    goto Again;
                }
            }
        }
        catch(Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
        finally
        {
            Console.ReadKey();
        }
    }
  }
}
```