Student.cs

```csharp
using System;


namespace OOP_Based_System
{
    public class Student
    {
        public string Name { get; set; }
        public string ClassSection { get; set; }
    }
}
```

Teacher.cs

```csharp
using System;


namespace OOP_Based_System
{
    public class Teacher
    {
        public string Name { get; set; }
        public string ClassSection { get; set; }
    }
}
```

Subject.cs

```csharp
using System;


namespace OOP_Based_System
{
    public class Subject
    {
        public string Name { get; set; }
        public string SubjectCode { get; set; }
        public Teacher Teacher { get; set; }
    }
}
```

Program.cs

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Remoting.Lifetime;
using System.Text;
using System.Threading.Tasks;

namespace OOP_Based_System
{
    public class Program
    {

        static List<Student> students = new List<Student>();
        static List<Teacher> teachers = new List<Teacher>();
        static List<Subject> subjects = new List<Subject>();
```

```csharp
        static void Main(string[] args)
        {// Sample data filling
            FillDummyData();

            // Displaying students in a class
            DisplayStudentsInClass("10 B");

            // Displaying subjects taught by a teacher
            DisplaySubjectsTaughtByTeacher("Sunil Gupta");

            Console.ReadKey();
        }

        static void FillDummyData()
        {
            // Adding dummy data for students
            students.Add(new Student { Name = "Aman", ClassSection = "10 A" });
            students.Add(new Student { Name = "Ashish", ClassSection = "10 B" });
            students.Add(new Student { Name = "Neha", ClassSection = "10 A" });
            students.Add(new Student { Name = "Sonu", ClassSection = "10 B" });
            students.Add(new Student { Name = "Ram", ClassSection = "10 A" });


            // Adding dummy data for teachers
            teachers.Add(new Teacher { Name = "Tarun Sharma", ClassSection = "10 A" });
            teachers.Add(new Teacher { Name = "Sunil Gupta", ClassSection = "10 B" });

            // Adding dummy data for subjects
            subjects.Add(new Subject { Name = "Mathematics", SubjectCode = "MATH101",
Teacher = teachers[0] });
            subjects.Add(new Subject { Name = "English", SubjectCode = "ENG101", Teacher
= teachers[1] });
            subjects.Add(new Subject { Name = "Science", SubjectCode = "SCI101", Teacher
= teachers[0] });
        }

        static void DisplayStudentsInClass(string classSection)
        {
            Console.WriteLine($"Students in class {classSection}:");
            foreach (var student in students)
            {
                if (student.ClassSection == classSection)
                    Console.WriteLine(student.Name);
            }
        }

        static void DisplaySubjectsTaughtByTeacher(string teacherName)
        {
            Console.WriteLine($"Subjects taught by {teacherName}:");
            foreach (var subject in subjects)
            {
                if (subject.Teacher.Name == teacherName)
                    Console.WriteLine($"{subject.Name} (Subject Code:
{subject.SubjectCode})");
            }
        }
    }
}
```