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

namespace ConAppStoringDataUsingOOPs
{
    public class Student
    {
        public string Name { get; set; }
        public string ClassAndSection { get; set; }
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConAppStoringDataUsingOOPs
{
    public class Teacher
    {
        public string Name { get; set; }
        public string ClassAndSection { get; set; }
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

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


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConAppStoringDataUsingOOPs
{
    public class SchoolManager
    {
        private List<Student> students = new List<Student>();
        private List<Teacher> teachers = new List<Teacher>();
        private List<Subject> subjects = new List<Subject>();
        public List<Teacher> Teachers => teachers;
        public void AddStudent(Student student)
        {
            students.Add(student);
        }
        public void AddTeacher(Teacher teacher)
        {
            teachers.Add(teacher);
        }
        public void AddSubject(Subject subject)
        {
            subjects.Add(subject);
        }
        public void DisplayStudentsInClass(string classAndSection)
        {
            Console.WriteLine($"Students in Class {classAndSection}");
            foreach(var student in students)
            {
                if (student.ClassAndSection == classAndSection)
                {
                    Console.WriteLine($"- {student.Name}");
                }
            }
        }
        public void DisplaySubjectsTaughtByTeacher(string teacherName)
        {
            Console.WriteLine($"Subjects taught by {teacherName}: ");
            foreach(var subject in subjects)
            {
                if(subject.Teacher.Name== teacherName)
                {
                    Console.WriteLine($"- {subject.Name} (Code:
{subject.SubjectCode}");
                }
            }
        }
    }
}


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConAppStoringDataUsingOOPs
{
    internal class Program
    {
        static void Main(string[] args)
        {
            SchoolManager schoolManager = new SchoolManager();
            bool exit = false;
            while (!exit)
            {
                Console.WriteLine("Select an option:");
                Console.WriteLine("1. Add Student");
                Console.WriteLine("2. Add Teacher");
                Console.WriteLine("3. Add Subject");
                Console.WriteLine("4. Display Students in a Class");
                Console.WriteLine("5. Display Subjects Taught by a Teacher");
                Console.WriteLine("6. Exit");

                string userInput = Console.ReadLine();
                switch (userInput)
                {
                    case "1":
                        Console.WriteLine("Enter Student Name:");
                        string studentName = Console.ReadLine();
                        Console.WriteLine("Enter Class and Section:");
                        string classAndSection = Console.ReadLine();
                        schoolManager.AddStudent(new Student { Name = studentName,
ClassAndSection = classAndSection });
                        break;

                    case "2":
                        Console.WriteLine("Enter Teacher Name:");
                        string teacherName = Console.ReadLine();
                        Console.WriteLine("Enter Class and Section:");
                        string teacherClassAndSection = Console.ReadLine();
                        schoolManager.AddTeacher(new Teacher { Name = teacherName,
ClassAndSection = teacherClassAndSection });
                        break;

                    case "3":
                        Console.WriteLine("Enter Subject Name:");
                        string subjectName = Console.ReadLine();
                        Console.WriteLine("Enter Subject Code:");
                        string subjectCode = Console.ReadLine();
                        Console.WriteLine("Select Teacher by Index (Enter the index
of the teacher from the displayed list):");
                        for (int i = 0; i < schoolManager.Teachers.Count; i++)
                        {
                            Console.WriteLine($"{i + 1}.
{schoolManager.Teachers[i].Name}");
                        }
                        int teacherIndex = int.Parse(Console.ReadLine()) - 1;
                        schoolManager.AddSubject(new Subject { Name = subjectName,
SubjectCode = subjectCode, Teacher = schoolManager.Teachers[teacherIndex] });
                        break;

                    case "4":
                        Console.WriteLine("Enter Class and Section to Display
Students:");

                        string classSectionToDisplay = Console.ReadLine();
                        schoolManager.DisplayStudentsInClass(classSectionToDisplay);
                        break;

                    case "5":
                        Console.WriteLine("Enter Teacher Name to Display 
Subjects:");
                        string teacherNameToDisplay = Console.ReadLine();

schoolManager.DisplaySubjectsTaughtByTeacher(teacherNameToDisplay);
                        break;

                    case "6":
                        exit = true;
                        break;

                    default:
                        Console.WriteLine("Invalid option. Please try again.");
                        break;
                }

                Console.WriteLine("Press any key to continue...");
                Console.ReadKey();
                Console.Clear();
            }
        }

    }
}
```