

Teacher.cs

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

namespace PhaseEndProject2
{
    public class Teacher
    {
        public int teacherId { get; set; }

        public string teacherName { get; set; }

        public string ClassSection { get; set; }

        public override string ToString()
        {
            return $"TeacherId:{teacherId},TeacherName:{teacherName},Class and
Section:{ClassSection}";
        }
    }
}
```

Program.cs

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

namespace PhaseEndProject2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string filePath = "teacher_data.txt";

            while (true)
            {
                Console.WriteLine("1. Add Teacher");
                Console.WriteLine("2. Update Teacher");
                Console.WriteLine("3. Exit");
            }
        }
    }
}
```

```

        Console.WriteLine("Enter your choice: ");
        int choice = int.Parse(Console.ReadLine());

        switch (choice)
        {
            case 1:
                AddTeacher(filePath);
                break;
            case 2:
                UpdateTeacher(filePath);
                break;
            case 3:
                Environment.Exit(0);
                break;
            default:
                Console.WriteLine("Invalid choice. Please try again.");
                break;
        }
    }

    static void AddTeacher(string filePath)
    {
        Console.WriteLine("Enter Teacher ID: ");
        int id = int.Parse(Console.ReadLine());
        Console.WriteLine("Enter Teacher Name: ");
        string name = Console.ReadLine();
        Console.WriteLine("Enter Class and Section: ");
        string classSection = Console.ReadLine();

        Teacher teacher = new Teacher { teacherId = id, teacherName = name,
ClassSection = classSection };

        using (StreamWriter writer = new StreamWriter(filePath, true))
        {
            writer.WriteLine($"{teacher.teacherId},{teacher.teacherName},{teacher.ClassSection}");
        }

        Console.WriteLine("Teacher data added successfully.");
    }

    static void UpdateTeacher(string filePath)
    {
        Console.WriteLine("Enter Teacher ID to update: ");
        int id = int.Parse(Console.ReadLine());

        string[] lines = File.ReadAllLines(filePath);

        bool found = false;

        using (StreamWriter writer = new StreamWriter(filePath, false))
        {
            foreach (string line in lines)
            {
                string[] parts = line.Split(',');
                if (parts.Length == 3)
                {
                    int currentId = int.Parse(parts[0]);
                    string currentName = parts[1];
                    string currentClassSection = parts[2];

```

```

        if (currentId == id)
        {
            Console.Write("Enter new Teacher Name: ");
            string newName = Console.ReadLine();
            Console.Write("Enter new Class and Section: ");
            string newClassSection = Console.ReadLine();

writer.WriteLine($"{currentId},{newName},{newClassSection}");
            found = true;
            Console.WriteLine("Teacher data updated
successfully.");
        }
        else
        {
            writer.WriteLine(line);
        }
    }
}

if (!found)
{
    Console.WriteLine("Teacher ID not found.");
}

Console.ReadLine();
}
}
}

```