SOURCE

Create a Text-file Based System For Storing and Updating Teacher Records

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq; // Added for LINQ

namespace StoringUpdatingTeacherRecords
{
    class Teacher
    {
        public int ID { get; set; }
        public string Name { get; set; }
        public string ClassSection { get; set; }

        public Teacher(int id, string name, string classSection)
        {
            ID = id;
            Name = name;
            ClassSection = classSection;
        }

        public override string ToString()
        {
            return $"{ID}, {Name}, {ClassSection}";
        }
    }

    class Program
    {
        static string filePath =
"C:\\Users\\Ragendhu\\source\\repos\\StoringUpdatingTeacherRecords\\teachers.txt";

        static void Main()
        {
            List<Teacher> teachers = ReadTeachersFromFile();

            while (true)
            {
                Console.WriteLine("1. Add new teacher");
                Console.WriteLine("2. Display all teachers");
                Console.WriteLine("3. Update teacher information");
```

```
        Console.WriteLine("4. Exit");
        Console.Write("Enter your choice: ");
        string choice = Console.ReadLine();

        switch (choice)
        {
            case "1":
                AddNewTeacher(teachers);
                break;
            case "2":
                DisplayAllTeachers(teachers);
                break;
            case "3":
                UpdateTeacherInformation(teachers);
                break;
            case "4":
                WriteTeachersToFile(teachers);
                Environment.Exit(0);
                break;
            default:
                Console.WriteLine("Invalid choice. Please try again.");
                break;
        }
    }
}

static List<Teacher> ReadTeachersFromFile()
{
    List<Teacher> teachers = new List<Teacher>();

    if (File.Exists(filePath))
    {
        string[] lines = File.ReadAllLines(filePath);
        foreach (string line in lines)
        {
            string[] parts = line.Split(',');
            int id = int.Parse(parts[0]);
            string name = parts[1].Trim();
            string classSection = parts[2].Trim();
            teachers.Add(new Teacher(id, name, classSection));
        }
    }

    return teachers;
```

```
}

static void WriteTeachersToFile(List<Teacher> teachers)
{
    List<string> lines = new List<string>();
    foreach (Teacher teacher in teachers)
    {
        lines.Add(teacher.ToString());
    }
    File.WriteAllLines(filePath, lines);
}

static void AddNewTeacher(List<Teacher> teachers)
{
    Console.Write("Enter teacher ID: ");
    int id = int.Parse(Console.ReadLine());

    Console.Write("Enter teacher name: ");
    string name = Console.ReadLine();

    Console.Write("Enter class and section: ");
    string classSection = Console.ReadLine();

    teachers.Add(new Teacher(id, name, classSection));
    Console.WriteLine("Teacher added successfully!");
}

static void DisplayAllTeachers(List<Teacher> teachers)
{
    Console.WriteLine("Teacher List (Sorted by ID ):");

    // Sort teachers by ID in ascending order using LINQ
    List<Teacher> sortedTeachers = teachers.OrderBy(t => t.ID).ToList();

    foreach (Teacher teacher in sortedTeachers)
    {
        Console.WriteLine(teacher);
    }
}

static void UpdateTeacherInformation(List<Teacher> teachers)
{
    Console.Write("Enter teacher ID to update: ");
    int idToUpdate = int.Parse(Console.ReadLine());
```

```
            Teacher teacherToUpdate = teachers.Find(t => t.ID == idToUpdate);

            if (teacherToUpdate != null)
            {
                Console.Write("Enter new name: ");
                teacherToUpdate.Name = Console.ReadLine();

                Console.Write("Enter new class and section: ");
                teacherToUpdate.ClassSection = Console.ReadLine();

                Console.WriteLine("Teacher information updated successfully!");
            }
            else
            {
                Console.WriteLine("Teacher not found with the given ID.");
            }
        }
    }

}
```