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

namespace MnPrj1
{
    class Teacher
    {
        public int ID { get; set; }
        public string Name { get; set; }
        public string ClassSection { get; set; }
    }

    class Program
    {
        static string filePath = "C:\\Mphs\\Mainproject1\\Student.txt";

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

            while (true)
            {
                Console.WriteLine("1. Add Teacher\n2. View Teachers\n3. Update 
Teacher\n4. Exit");
                int choice = int.Parse(Console.ReadLine());

                switch (choice)
                {
                    case 1:
                        AddTeacher(teachers);
                        break;
                    case 2:
                        ViewTeachers(teachers);
                        break;
                    case 3:
                        UpdateTeacher(teachers);
                        break;
                    case 4:
                        SaveData(teachers);
                        Environment.Exit(0);
                        break;
                    default:
                        Console.WriteLine("Invalid choice. Please try again.");
                        break;
                }
            }
        }

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

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

                foreach (var line in lines)
                {
                    string[] data = line.Split(',');
                    teachers.Add(new Teacher { ID = int.Parse(data[0]), Name = 
data[1], ClassSection = data[2] });
                }
            }

            return teachers;
        }

        static void SaveData(List<Teacher> teachers)
        {
            using (StreamWriter writer = new StreamWriter(filePath))
            {
                foreach (var teacher in teachers)
                {

writer.WriteLine($"{teacher.ID},{teacher.Name},{teacher.ClassSection}");
                }
            }
        }

        static void AddTeacher(List<Teacher> teachers)
        {
            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();

            teachers.Add(new Teacher { ID = id, Name = name, ClassSection = 
classSection });

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

        static void ViewTeachers(List<Teacher> teachers)
        {
            Console.WriteLine("Teacher List:");
            foreach (var teacher in teachers)
            {
                Console.WriteLine($"ID: {teacher.ID}, Name: {teacher.Name}, Class 
and Section: {teacher.ClassSection}");
            }
        }

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

            Teacher teacherToUpdate = teachers.Find(t => t.ID == idToUpdate);
            if (teacherToUpdate != null)
            {
                Console.WriteLine($"Current details: ID: {teacherToUpdate.ID}, Name: 
{teacherToUpdate.Name}, Class and Section: {teacherToUpdate.ClassSection}");

                Console.WriteLine("Enter new name:");
                teacherToUpdate.Name = Console.ReadLine();

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

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

}
```