**WriteUp for Modified Code:**

GitHub Link: <https://github.com/niljames/simplilearn.git>

In my other code, I have included only two functions display() and add(). Please consider the new code.

This is the new code which also includes the updating the already existing teacher’s records.

**.txt File: (before Update)**

![Graphical user interface, application, Word

Description automatically generated]()

**Code:**

using System;

using System.Collections.Generic;

using System.IO;

namespace Assessment1

{

class Program

{

public static List<int> assign()

{

var File\_name = @"D:\Assignment for Simplilearn\Assignment 1\Teacher\_Details.txt";

string[] lines = File.ReadAllLines(File\_name);

var id = new List<int>();

int count = 5;

foreach (string line in lines)

{

var data = line.Split(',');

id.Add(Convert.ToInt32(data[0]));

count++;

}

return id;

}

public static int display()

{

var File\_name = @"D:\Assignment for Simplilearn\Assignment 1\Teacher\_Details.txt";

string[] lines = File.ReadAllLines(File\_name);

var Name = new List<string>();

var id = new List<int>();

var clss = new List<string>();

var Section = new List<string>();

int count = 0;

foreach (string line in lines)

{

var data = line.Split(',');

id.Add(Convert.ToInt32(data[0]));

Name.Add(data[1]);

clss.Add(data[2]);

Section.Add(data[3]);

count++;

}

Console.WriteLine("Content of the Teacher\_Details.txt file:");

Console.ForegroundColor = ConsoleColor.DarkYellow;

Console.WriteLine("Id\t\tName\t\tClass\t\tSection");

Console.ForegroundColor = ConsoleColor.White;

for (int i = 0; i < count; i++)

{

Console.WriteLine(id[i] + "\t\t" + Name[i] + "\t\t" + clss[i] + "\t\t" + Section[i]);

}

return count;

}

public static void add(int count)

{

Console.WriteLine("Enter the details to be updated:");

Console.Write("Id: ");

int Id = Convert.ToInt32(Console.ReadLine());

List<int> id1 = new List<int>();

id1=assign();

for(int j = 0; j < count; j++)

{

if (Id==id1[j])

{

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("Cannot add as no two teachers can have the same ids.");

Console.ForegroundColor = ConsoleColor.White;

return;

}

}

Console.Write("Name: ");

string Name = Console.ReadLine();

Console.Write("Class: ");

string clss = Console.ReadLine();

Console.Write("Section: ");

string Section = Console.ReadLine();

var File\_name = @"D:\Assignment for Simplilearn\Assignment 1\Teacher\_Details.txt";

using (StreamWriter sw = File.AppendText(File\_name))

{

sw.WriteLine(Id + "," + Name + "," + clss + "," + Section);

}

Console.WriteLine("\nDone Adding...");

}

public static void update(int id, int count)

{

var File\_name = @"D:\Assignment for Simplilearn\Assignment 1\Teacher\_Details.txt";

string[] lines = File.ReadAllLines(File\_name);

var id1 = new List<int>();

var Name = new List<string>();

var clss = new List<string>();

var Section = new List<string>();

foreach (string line in lines)

{

var data = line.Split(',');

id1.Add(Convert.ToInt32(data[0]));

Name.Add(data[1]);

clss.Add(data[2]);

Section.Add(data[3]);

count++;

}

string N,s,C;

for (int j = 0; j < count; j++)

{

if (id == id1[j])

{

N = Name[j];

Console.Write("Enter the New Name to be updated: ");

Name[j] = Console.ReadLine();

C = clss[j];

Console.Write("Enter the New Class to be updated: ");

clss[j] = Console.ReadLine();

Console.Write("Enter the New Section to be updated: ");

s = Section[j];

Section[j] = Console.ReadLine();

String[] str = File.ReadAllLines(File\_name);

for (int i = 0; i < str.Length; i++) {

if (str[i].Contains(N))

{

str[i] = str[i].Replace(N, Name[j]);

str[i] = str[i].Replace(C, clss[j]);

str[i] = str[i].Replace(s, Section[j]);

}

}

File.WriteAllLines(File\_name,str);

Console.WriteLine("\nDone Updating..........");

return;

}

}

Console.WriteLine("No id found.");

}

static void Main(string[] args)

{

int count = 0;

Console.ForegroundColor = ConsoleColor.DarkMagenta;

Console.BackgroundColor = ConsoleColor.Yellow;

Console.WriteLine("\nWelcome to the Teacher's Database.");

Console.ForegroundColor = ConsoleColor.White;

Console.BackgroundColor = ConsoleColor.Black;

string choice = "Y";

do

{

Console.WriteLine("What functionality would you like to perform?");

Console.WriteLine("\n\t1.Retrieve and Display\n\t2.Add\n\t3.Update\n");

Console.WriteLine("Enter the option:");

int option = Convert.ToInt32(Console.ReadLine());

switch (option)

{

case 1:

count = display();

break;

case 2:

add(count);

break;

case 3:

Console.Write("Enter Id of the record that you want to update:");

int id = Convert.ToInt32(Console.ReadLine());

update(id,count);

break;

default:

Console.WriteLine("EXITING.....");

break;

}

Console.WriteLine("\nDo you want to continue? (Y/N)");

choice = Console.ReadLine();

Console.WriteLine();

} while (choice.ToUpper() == "Y");

Console.ForegroundColor = ConsoleColor.DarkMagenta;

Console.BackgroundColor = ConsoleColor.Yellow;

Console.WriteLine("\nThank you for using the Teacher's Database.");

Console.ReadKey();

Console.ForegroundColor = ConsoleColor.White;

Console.BackgroundColor = ConsoleColor.Black;

}

}

}

**Explanation of the code:**

There are three functions:

* Display()
* Add()
* Update()

The Display() function is used to display all the lines of the .txt file in a tabular format that is arranged neatly. Uses a simple for loop to go through all the lines and print them in the console.

The Add() function has another function called assign() function which checks, if the new record to be added already exist. As in..it doesn’t allow duplicate ids to be added. If no issues, it goes ahead and append it to the end of the file with the help of StreamWriter.

The Update() Function is used to update an already existing record. This is used to make changes of a particular record. And if done successfully, the changes will be updated in the txt file.

All the three functions are called via a switch() statement in the Main function where the user enters the option of the functionality that he/she wants to perform.

**Output:**

![Text

Description automatically generated]()

![Text

Description automatically generated]()![Text

Description automatically generated]()![Text

Description automatically generated]()![Text

Description automatically generated]()

**.txt file after update:**

