Shatha Salem Alkulaib

Project: Create a Text-file Based System for Storing and Updating Teacher Records

Program: Full Stack Developer .NET

Phase: one

GitHub link: https://github.com/ShathaAlk/FinalProject_CSharp

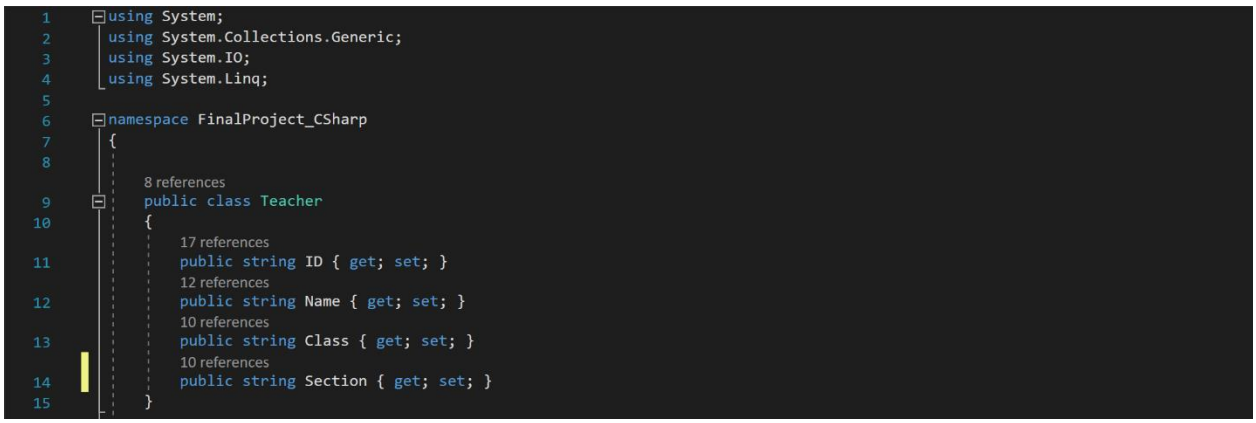
Email: shatha.alkulaib@gmail.com

The program consists of six main operations:

1- Create a File in a specific folder 2- Store Data to file 3- Sort the data

4- Search about data 5- Update the data 6- Retrieve the data from the file.

In the beginning, creating a Teacher Class that contains four variables.



```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

namespace FinalProject_CSharp
{

    8 references
    public class Teacher
    {
        17 references
        public string ID { get; set; }
        12 references
        public string Name { get; set; }
        10 references
        public string Class { get; set; }
        10 references
        public string Section { get; set; }
    }
```

Then, Creating a FileHandlingOperations Class that contains three methods, WriteData(), StoreData() and ReadData(), they will be explained in more detail in the next pages.

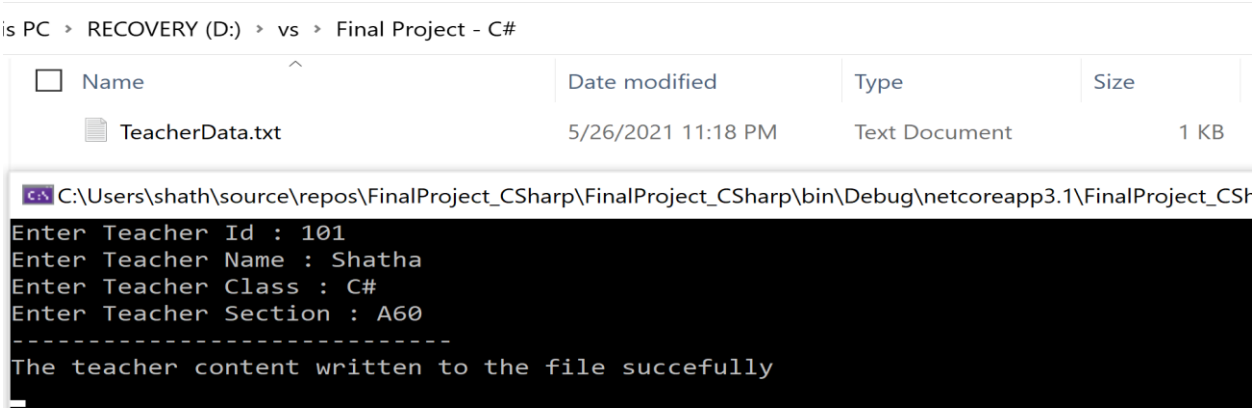
```
    0 references
    class Program
    {
        2 references
        class FileHandlingOperations
        {
            1 reference
            public void WriteData()...
            1 reference
            public void StoreData()...
            1 reference
            public void ReadData()...
        }
```

The program starts with the IF condition that if the file is not in the folder, it will be created by using the WriteData() method. It used for once to create a file and add a first teacher record in it.

```
static void Main(string[] args)
{
    {
        FileHandlingOperations FOP = new FileHandlingOperations();
        string path = "D:\\vs\\Final Project - C#\\TeacherData.txt";
        if (!File.Exists(path))
        {
            FOP.WriteData();
        }
        else
        {
```

```
1 reference
public void WriteData()
{
    FileStream fs = new FileStream("D:\\vs\\Final Project - C#\\TeacherData.txt", FileMode.Create, FileAccess.Write);
    StreamWriter sw = new StreamWriter(fs);

    Teacher teacher = new Teacher();
    Console.Write("Enter Teacher Id : ");
    teacher.ID = Console.ReadLine();
    Console.Write("Enter Teacher Name : ");
    teacher.Name = Console.ReadLine();
    Console.Write("Enter Teacher Class : ");
    teacher.Class = Console.ReadLine();
    Console.Write("Enter Teacher Section : ");
    teacher.Section = Console.ReadLine();
    Console.WriteLine("----------------------------");

    string teacherData1 = teacher.ID + "," + teacher.Name + "," + teacher.Class + "," + teacher.Section;
    sw.WriteLine(teacherData1);

    Console.WriteLine("The teacher content written to the file succefully");
    sw.Close();
    fs.Close();
}
```

Output:



After creating the file and adding the first record for the teacher, it will start executing the else condition, which is the rest of the five main operations inside the main switch case according to the user's use.

```
else
{
    char MainOpt;
    try
    {
        do
        {
            //Declaring the text file, then covert it into array.
            string fileData = File.ReadAllText("D:\\vs\\Final Project - C#\\TeacherData.txt");
            string[] arrData = fileData.Split("".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);

            //Split the data to four variables to deal with them individually.
            List<Teacher> teacherList = new List<Teacher>();
            foreach (string teacherRecord in arrData)
            {
                Teacher teacher = new Teacher();
                string[] splitData = teacherRecord.Split(',');
                teacher.ID = splitData[0];
                teacher.Name = splitData[1];
                teacher.Class = splitData[2];
                teacher.Section = splitData[3];
                teacherList.Add(teacher);
            }
            //The five main processes.
            Console.WriteLine("1 - Store Teacher Data");
            Console.WriteLine("2 - Sort Teacher Records");
            Console.WriteLine("3 - Search about Teacher Information");
            Console.WriteLine("4 - Update Teacher Records");
            Console.WriteLine("5 - Retrieve Teacher Data");
            Console.WriteLine("Please Select Your Main Choice : ");
```

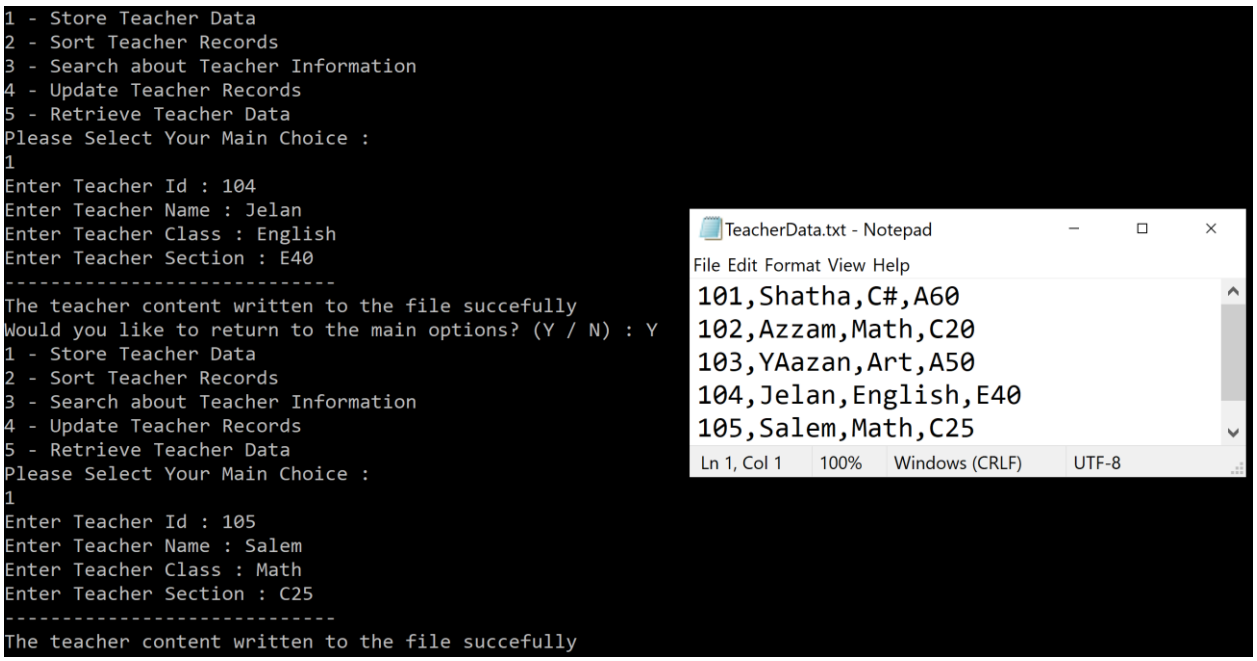
The first case in the main switch case is Storing data by using StoreData() method. It used to append many teacher records to the TeacherData.txt file.

```
int MainChoice = int.Parse(Console.ReadLine());
switch (MainChoice)
{
    case 1:
        FOP.StoreData();
        break;
```

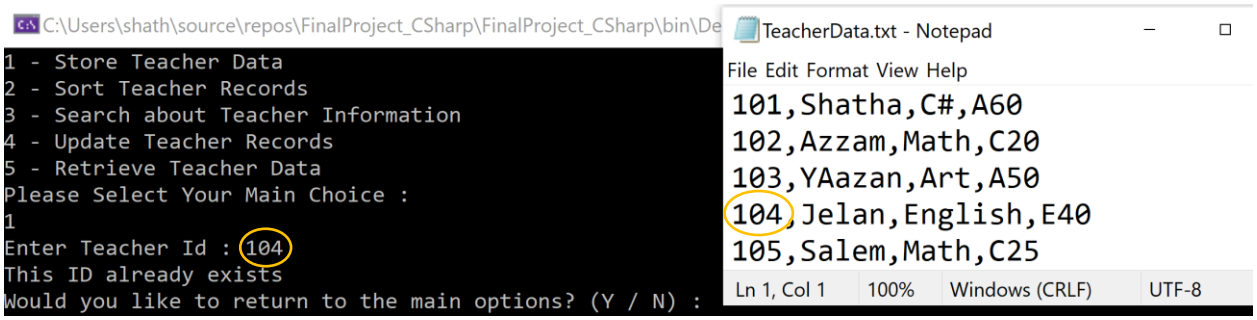
```
1 reference
public void StoreData()
{
    try
    {
        string fileData = File.ReadAllText("D:\\vs\\Final Project - C#\\TeacherData.txt");

        //Store data to text file.
        using (FileStream fs = new FileStream("D:\\vs\\Final Project - C#\\TeacherData.txt", FileMode.Append, FileAccess.Write))
        {
            using (StreamWriter sw = new StreamWriter(fs))
            {
                // Add the variables' values to the text file.
                Teacher teacher = new Teacher();
                Console.Write("Enter Teacher Id : ");
                teacher.ID = Console.ReadLine();
                if (fileData.Contains(teacher.ID))
                {
                    Console.WriteLine("This ID already exists");
                }
                else
                {
                    Console.Write("Enter Teacher Name : ");
                    teacher.Name = Console.ReadLine();
                    Console.Write("Enter Teacher Class : ");
                    teacher.Class = Console.ReadLine();
                    Console.Write("Enter Teacher Section : ");
                    teacher.Section = Console.ReadLine();
                    Console.WriteLine("----------------------------");
                    string teacherData1 = teacher.ID + "," + teacher.Name + "," + teacher.Class + "," + teacher.Section;
                    sw.WriteLine(teacherData1);
                    Console.WriteLine("The teacher content written to the file succefully");
                    sw.Close();
                    fs.Close();
                }
            }
        }
    }
```

```
    catch (IOException ex)
    {
        Console.WriteLine(ex.Message);
    }
}
1 reference
```

Output: Adding more four teachers.



If the ID Already exists, a validation message will be shown.



The second case in the main switch case is Sorting records either by teacher ID or name, a sub-switch case was used for both cases of sorting.

```
case 2:
    char opt;
    do
    {
        // Sorting Process
        Console.WriteLine("1 - Sorting by ID");
        Console.WriteLine("2 - Sorting by Name");
        Console.WriteLine("Please Select Your Sort Choice: ");
        int choice = int.Parse(Console.ReadLine());

        switch (choice)
        {
            case 1:
                //Sort the records depend on teacher ID.
                var sortedlistID = teacherList.OrderBy(s => s.ID);

                foreach (var teacherRec in sortedlistID)
                {
                    Console.WriteLine(string.Format("{0} , {1} , {2} , {3}", teacherRec.ID, teacherRec.Name, teacherRec.Class, teacherRec.Section));
                }
                break;

            case 2:
                //Sort the records depend on teacher Name
                var sortedlistName = teacherList.OrderBy(s => s.Name);
                foreach (var teacherRec in sortedlistName)
                {
                    Console.WriteLine(string.Format("{0} , {1} , {2} , {3}", teacherRec.Name, teacherRec.ID, teacherRec.Class, teacherRec.Section));
                }
                break;

            default:
                Console.WriteLine("Invalid Choice!!");
                break;
        }
        Console.Write("Do you wish to Continue Sorting? (Y / N) : ");
        opt = char.Parse(Console.ReadLine());
    } while (opt == 'y' || opt == 'Y');
    break;
```

Output:

```
1 - Store Teacher Data
2 - Sort Teacher Records
3 - Search about Teacher Information
4 - Update Teacher Records
5 - Retrieve Teacher Data
Please Select Your Main Choice :
2
1 - Sorting by ID
2 - Sorting by Name
Please Select Your Sort Choice:
1
101 , Shatha , C# , A60
102 , Azzam , Math , C20
103 , YAazan , Art , A50
104 , Jelan , English , E40
105 , Salem , Math , C25
Do you wish to Continue Sorting? (Y / N) : y
1 - Sorting by ID
2 - Sorting by Name
Please Select Your Sort Choice:
2
Azzam , 102 , Math , C20
Jelan , 104 , English , E40
Salem , 105 , Math , C25
Shatha , 101 , C# , A60
YAazan , 103 , Art , A50
Do you wish to Continue Sorting? (Y / N) :
```

Sorting by ID

Sorting by name

The third case in the main switch case is Searching about specific records either by teacher ID or name, a sub-switch case was used for both cases of sorting.

```
case 3:
    do
    {
        Console.WriteLine("1 - Searching by Teacher ID");
        Console.WriteLine("2 - Searching by Teacher Name");
        Console.WriteLine("Please Select Your Search choice : ");
        int choice = int.Parse(Console.ReadLine());

        switch (choice)
        {
            case 1:
                Console.WriteLine("Enter Teacher ID to search:");
                string searchID = Console.ReadLine();

                //Search by Teacher ID
                foreach (var teacherRec in teacherList)
                {
                    //Searching process depend on teacher ID
                    if (teacherRec.ID == searchID)
                    {
                        Console.WriteLine(string.Format("{0} , {1} , {2} , {3}", teacherRec.ID, teacherRec.Name, teacherRec.Class, teacherRec.Section));
                    }
                    if (!fileData.Contains(searchID))
                    {
                        Console.WriteLine("Not Found");
                        break;
                    }
                }
                break;
```

```
            case 2:
                Console.WriteLine("Enter Teacher Name to Search:");
                string searchName = Console.ReadLine();

                //Searching process depend on teacher Name.
                foreach (var teacherRec in teacherList)
                {
                    if (teacherRec.Name.Contains(searchName))
                    {

                        Console.WriteLine(string.Format("{0} , {1} , {2} , {3}", teacherRec.Name, teacherRec.ID, teacherRec.Class, teacherRec.Section));
                    }

                    if (!fileData.Contains(searchName))
                    {
                        Console.WriteLine("Not Found");
                        break;

                    }
                }

                break;

            default:
                Console.WriteLine("Invalid Choice!!");
                break;
        }
        Console.Write("Do you wish to Continue Searching? (Y / N) : ");
        opt = char.Parse(Console.ReadLine());
    } while (opt == 'y' || opt == 'Y');

    break;
```

Output:

```
1 - Store Teacher Data
2 - Sort Teacher Records
3 - Search about Teacher Information
4 - Update Teacher Records
5 - Retrieve Teacher Data
Please Select Your Main Choice :
3
1 - Searching by Teacher ID
2 - Searching by Teacher Name
Please Select Your Search choice :
1
Enter Teacher ID to search:
105
105 , Salem , Math , C25
```

Searching by ID

```
1 - Searching by Teacher ID
2 - Searching by Teacher Name
Please Select Your Search choice :
2
Enter Teacher Name to Search:
Jelan
Jelan , 104 , English , E40
```

Searching by name

If the ID or name does not exist, a validation message will be shown.

```
Enter Teacher ID to search:
109
Not Found
Do you wish to Continue Searching? (Y / N) : y
1 - Searching by Teacher ID
2 - Searching by Teacher Name
Please Select Your Search choice :
2
Enter Teacher Name to Search:
Omar
Not Found
```

TeacherData.txt - Notepad

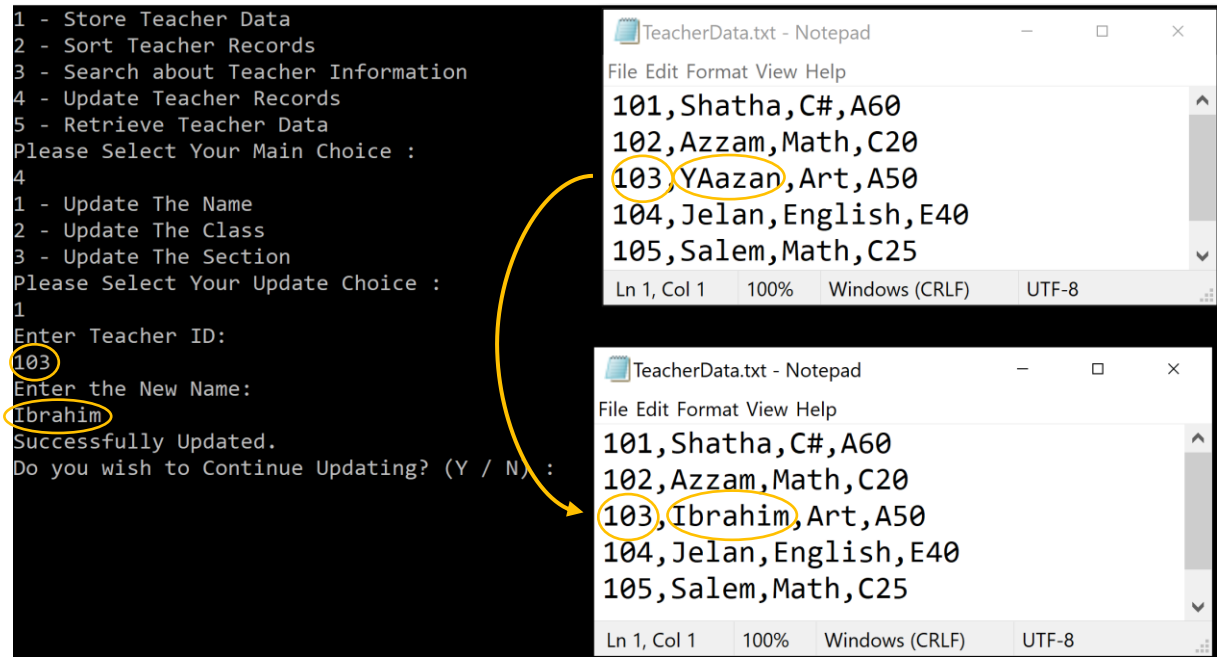
File Edit Format View Help

```
101,Shatha,C#,A60
102,Azzam,Math,C20
103,YAazan,Art,A50
104,Jelan,English,E40
105,Salem,Math,C25
```

The fourth case is Updating specific records by using teacher ID, a sub-switch case was used for three update cases which are Update Name, Update Class, and Update Section.

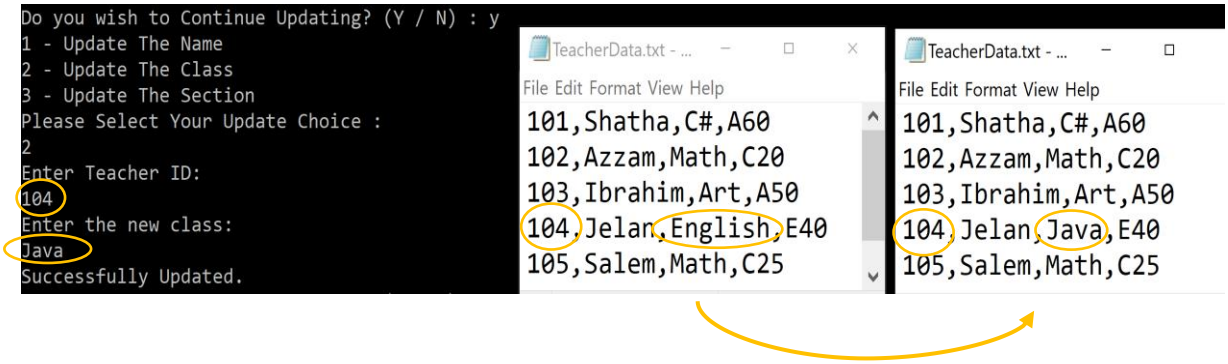
```
case 4:
    do
    {
        //Redeclaring the file to get the last updated version of the file.
        string fileData2 = File.ReadAllText("D:\\vs\\Final Project - C#\\TeacherData.txt");
        Console.WriteLine("1 - Update The Name");
        Console.WriteLine("2 - Update The Class");
        Console.WriteLine("3 - Update The Section");
        Console.WriteLine("Please Select Your Update Choice : ");
        int choice = int.Parse(Console.ReadLine());

        switch (choice)
        {
            case 1:
                Console.WriteLine("Enter Teacher ID:");
                string TID1 = Console.ReadLine();

                //Updating process by using ID
                foreach (var teacherRec in teacherList)
                {
                    if (teacherRec.ID == TID1)
                    {
                        //Update teacher name.
                        Console.WriteLine("Enter the New Name:");
                        string newName = Console.ReadLine();
                        //Use Replace() method  to change the record.
                        string newtext = fileData2.Replace(teacherRec.Name, newName);
                        File.WriteAllText("D:\\vs\\Final Project - C#\\TeacherData.txt", newtext);
                        Console.WriteLine("Successfully Updated.");
                    }
                    if (!fileData2.Contains(TID1))
                    {
                        Console.WriteLine("Not Found");
                        break;
                    }
                }

                break;
```

Output - Case 1 (Update Name):



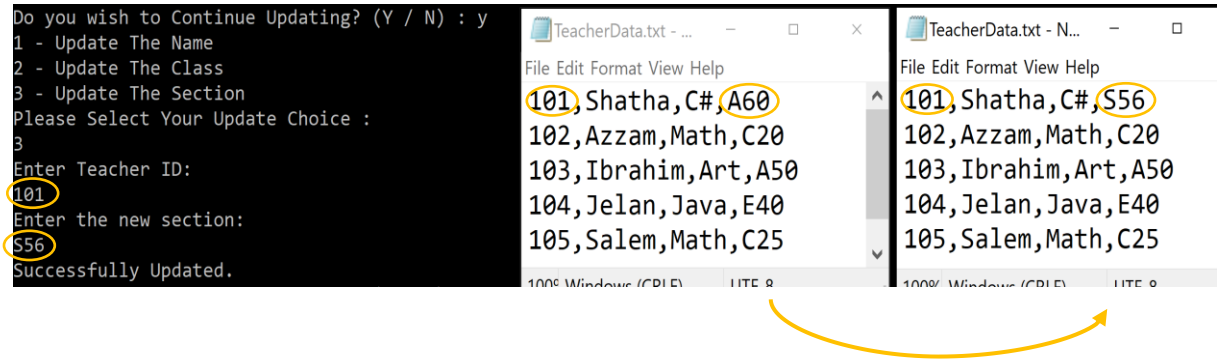

```
case 2:
    Console.WriteLine("Enter Teacher ID:");
    string TID2 = Console.ReadLine();

    //Update teacher records, Then save the results in the same file.
    foreach (var teacherRec in teacherList)
    {
        if (teacherRec.ID == TID2)
        {
            //Update teacher class.
            Console.WriteLine("Enter the new class:");
            string newClass = Console.ReadLine();

            string newtext = fileData2.Replace(teacherRec.Class, newClass);
            File.WriteAllText("D:\\vs\\Final Project - C#\\TeacherData.txt", newtext);
            Console.WriteLine("Successfully Updated.");
        }
        if (!fileData2.Contains(TID2))
        {
            Console.WriteLine("Not Found");
            break;
        }
    }

    break;
```

Output - Case 2 (Update Class):



```
        case 3:
            Console.WriteLine("Enter Teacher ID:");
            string TID3 = Console.ReadLine();

            foreach (var teacherRec in teacherList)
            {
                if (teacherRec.ID == TID3)
                {
                    //Update teacher section.
                    Console.WriteLine("Enter the new section:");
                    string newSection = Console.ReadLine();

                    string newtext = fileData2.Replace(teacherRec.Section, newSection);
                    File.WriteAllText("D:\\vs\\Final Project - C#\\TeacherData.txt", newtext);
                    Console.WriteLine("Successfully Updated.");
                }

                if (!fileData2.Contains(TID3))
                {
                    Console.WriteLine("Not Found");
                    break;

                }
            }

            break;

        default:
            Console.WriteLine("Invalid Choice!!");
            break;
    }

    Console.Write("Do you wish to Continue Updating? (Y / N) : ");
    opt = char.Parse(Console.ReadLine());
} while (opt == 'y' || opt == 'Y');

break;
```

Output - Case 3 (Update Section):



If the ID does not exist, a validation message will be shown.

The fifth case is Retrieving data from the TeacherData.txt file by using ReadData() method which used to retrieve the file contents and show them in the Console Application.

```
            case 5:
                FOP.ReadData();
                break;


            default:
                Console.WriteLine("Invalid Choice!!");
                break;
        }
        Console.Write("Would you like to return to the main options? (Y / N) : ");
        MainOpt = char.Parse(Console.ReadLine());
    } while (MainOpt == 'y' || MainOpt == 'Y');
```

```
1 reference
public void ReadData()
{
    FileStream fs = new FileStream("D:\\vs\\Final Project - C#\\TeacherData.txt", FileMode.Open, FileAccess.Read);
    StreamReader sr = new StreamReader(fs);
    sr.BaseStream.Seek(0, SeekOrigin.Begin);
    string str = sr.ReadLine();
    while (str != null)
    {
        Console.WriteLine(str);
        str = sr.ReadLine();
    }
    sr.Close();
    fs.Close();
}
```

Output:

```
1 - Store Teacher Data
2 - Sort Teacher Records
3 - Search about Teacher Information
4 - Update Teacher Records
5 - Retrieve Teacher Data
Please Select Your Main Choice :
5
101,Shatha,C#,S56
102,Azzam,Math,C20
103,Ibrahim,Art,A50
104,Jelan,Java,E40
105,Salem,Math,C25
```

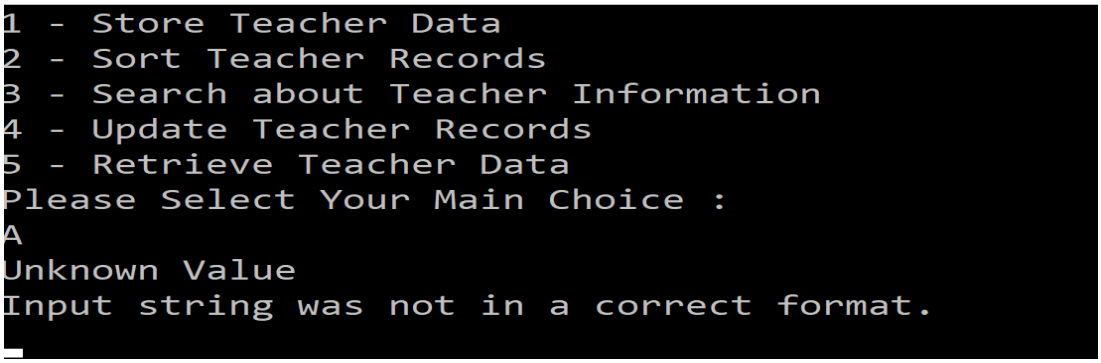
TeacherData.txt - Notepad

File Edit Format View Help

```
101,Shatha,C#,S56
102,Azzam,Math,C20
103,Ibrahim,Art,A50
104,Jelan,Java,E40
105,Salem,Math,C25
```

Exception Handling is implemented by using Try-catch to solve the problem of entering wrong values in the console application.

```
1 - Store Teacher Data
2 - Sort Teacher Records
3 - Search about Teacher Information
4 - Update Teacher Records
5 - Retrieve Teacher Data
Please Select Your Main Choice :
6
Invalid Choice!!
Would you like to return to the main options? (Y / N) : yes
Unknown Value
String must be exactly one character long.
```


1 - Store Teacher Data
2 - Sort Teacher Records
3 - Search about Teacher Information
4 - Update Teacher Records
5 - Retrieve Teacher Data
Please Select Your Main Choice :
A
Unknown Value
Input string was not in a correct format.