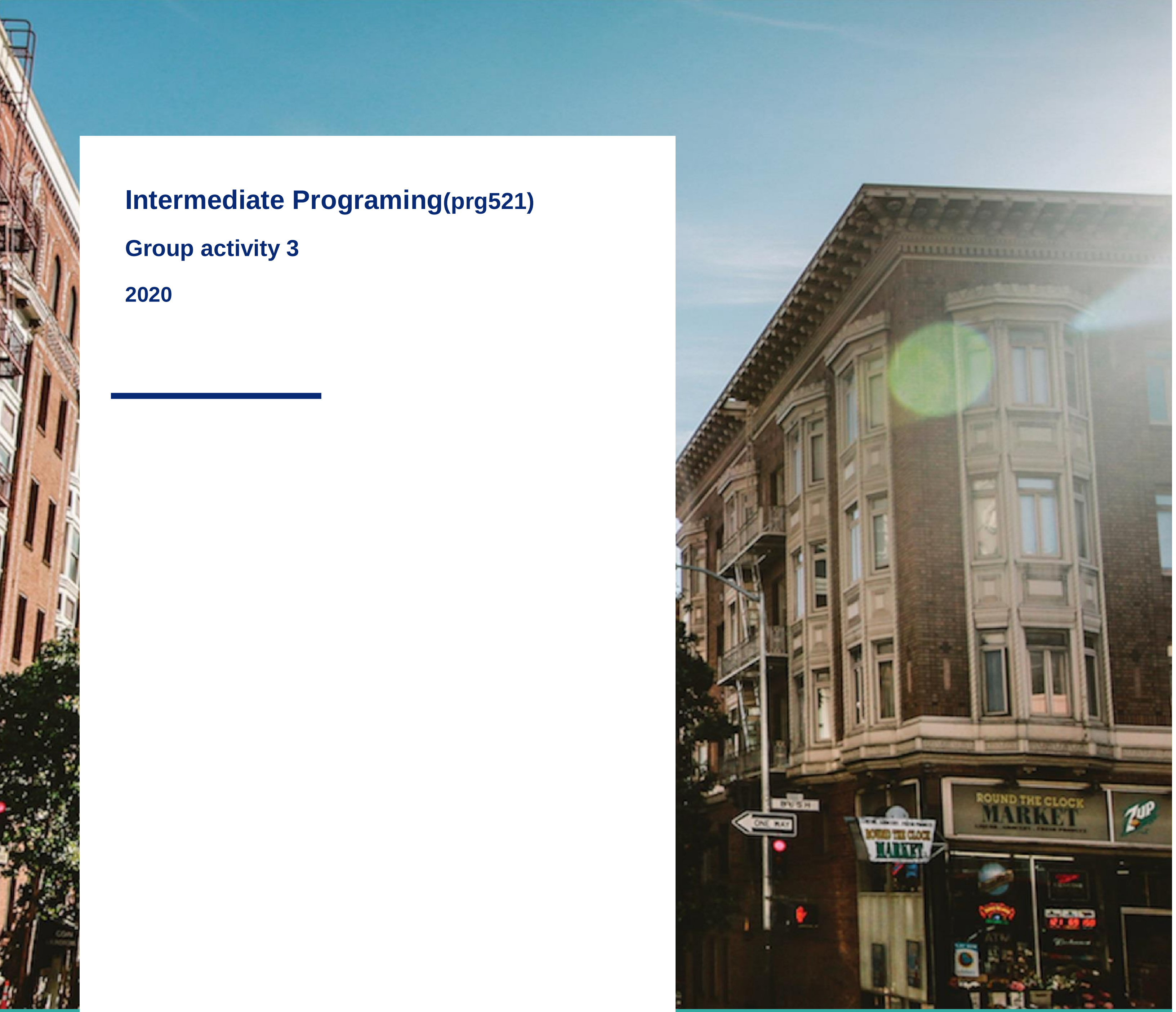


Intermediate Programing(prg521)

Group activity 3

2020

APRIL 10

Ctu Training Solutions
Authored by: Albert Michael Ludick

Logo
Name

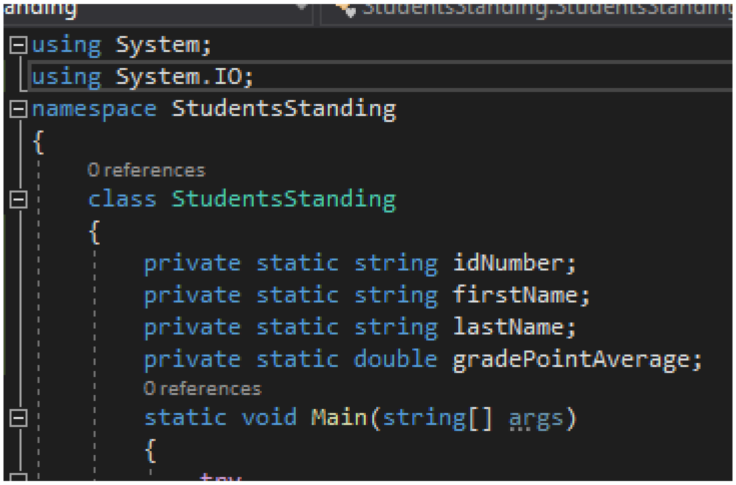
Table of Contents

Program Content

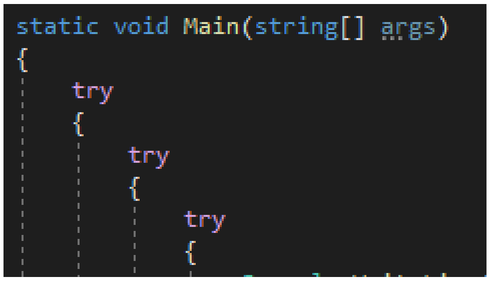
Source Code:

Question 1

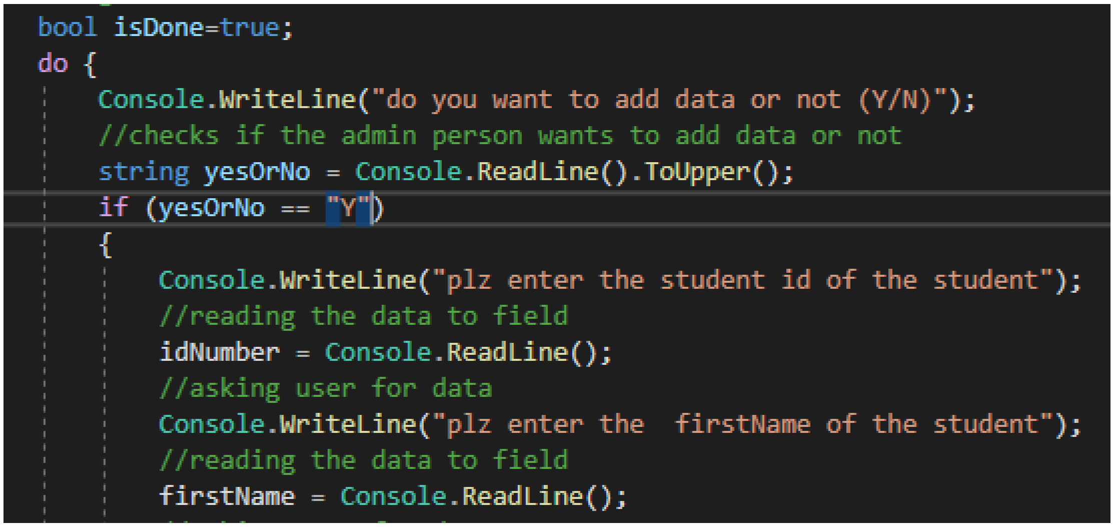
class StudentsStanding:



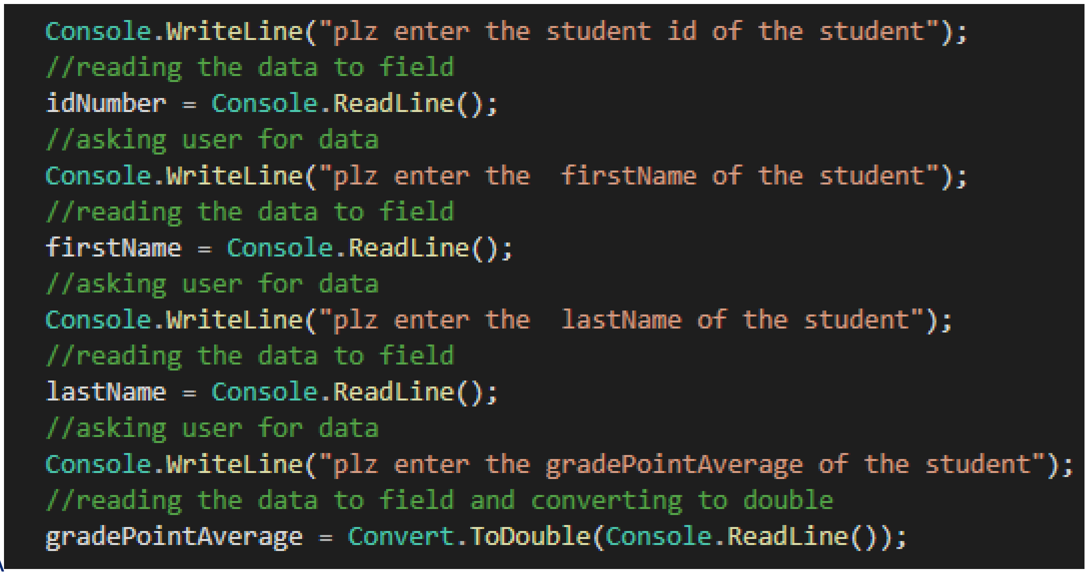
```
using System;
using System.IO;
namespace StudentsStanding
{
    0 references
    class StudentsStanding
    {
        private static string idNumber;
        private static string firstName;
        private static string lastName;
        private static double gradePointAverage;
        0 references
        static void Main(string[] args)
        {
```



```
static void Main(string[] args)
{
    try
    {
        try
        {
            try
            {
```

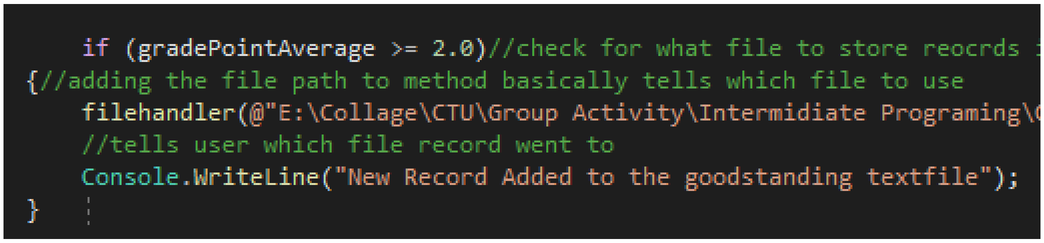


```
bool isDone=true;
do {
    Console.WriteLine("do you want to add data or not (Y/N)");
    //checks if the admin person wants to add data or not
    string yesOrNo = Console.ReadLine().ToUpper();
    if (yesOrNo == "Y")
    {
        Console.WriteLine("plz enter the student id of the student");
        //reading the data to field
        idNumber = Console.ReadLine();
        //asking user for data
        Console.WriteLine("plz enter the  firstName of the student");
        //reading the data to field
        firstName = Console.ReadLine();
```

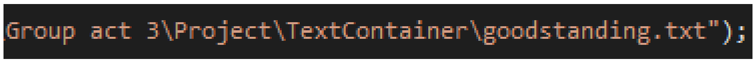


```
Console.WriteLine("plz enter the student id of the student");
//reading the data to field
idNumber = Console.ReadLine();
//asking user for data
Console.WriteLine("plz enter the  firstName of the student");
//reading the data to field
firstName = Console.ReadLine();
//asking user for data
Console.WriteLine("plz enter the  lastName of the student");
//reading the data to field
lastName = Console.ReadLine();
//asking user for data
Console.WriteLine("plz enter the gradePointAverage of the student");
//reading the data to field and converting to double
gradePointAverage = Convert.ToDouble(Console.ReadLine());
```

\

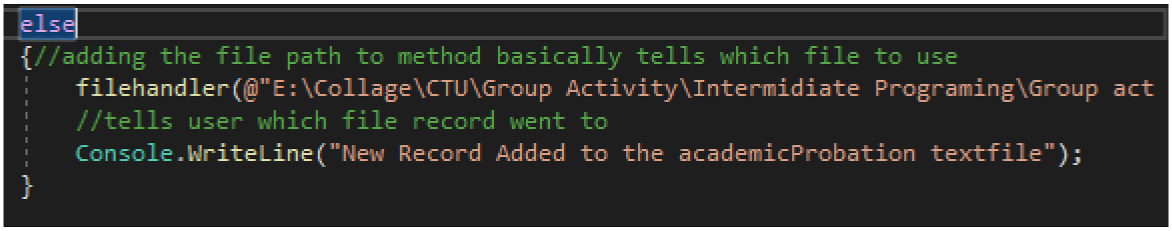


```
    if (gradePointAverage >= 2.0)//check for what file to store reocrds
 {//adding the file path to method basically tells which file to use
    filehandler(@"E:\Collage\CTU\Group Activity\Intermidiate Programing\
    //tells user which file record went to
    Console.WriteLine("New Record Added to the goodstanding textfile");
 }
```

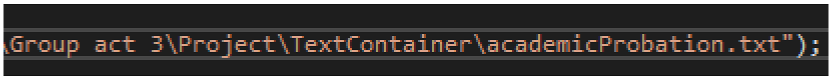


```
Group act 3\Project\TextContainer\goodstanding.txt");
```

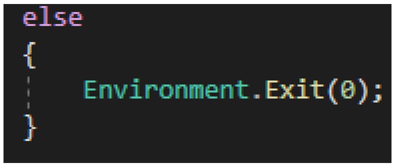
@"E:\Collage\CTU\Group Activity\Intermidiate Programing\Group act 3\Project\TextContainer\goodstanding.txt" is what is passed to the fileHandler method in the if statement as its path argument



```
else
{//adding the file path to method basically tells which file to use
    filehandler(@"E:\Collage\CTU\Group Activity\Intermidiate Programing\Group act
    //tells user which file record went to
    Console.WriteLine("New Record Added to the academicProbation textfile");
}
```

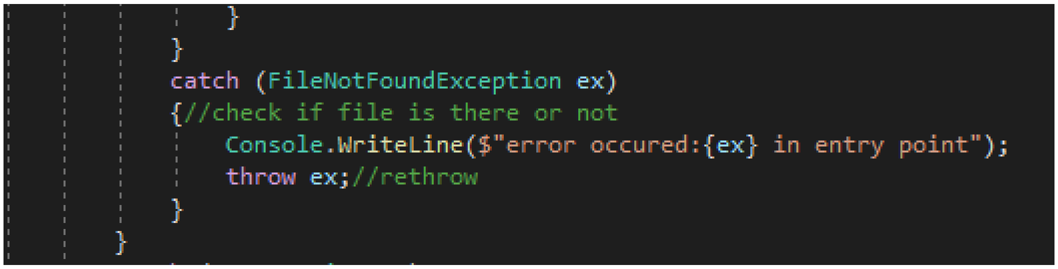


```
\Group act 3\Project\TextContainer\academicProbation.txt");
```

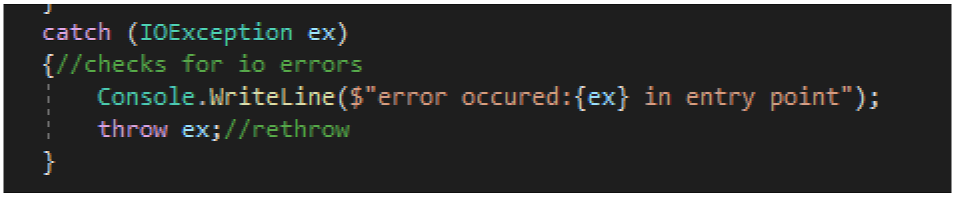


```
else
{
    Environment.Exit(0);
}
```

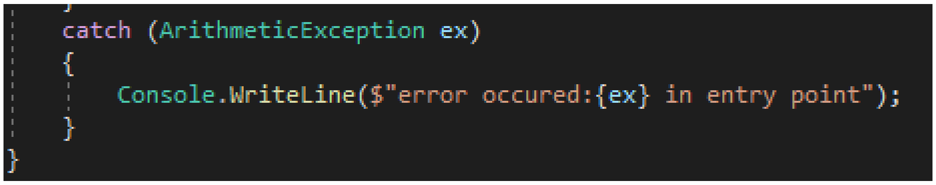
@"E:\Collage\CTU\Group Activity\Intermidiate Programing\Group act 3\Project\TextContainer\academicProbation.txt" is what is passed to the fileHandler method in the else body as its path argument



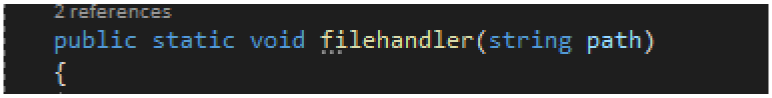
```
                }
            }
            catch (FileNotFoundException ex)
            {//check if file is there or not
                Console.WriteLine($"error occured:{ex} in entry point");
                throw ex;//rethrow
            }
        }
```



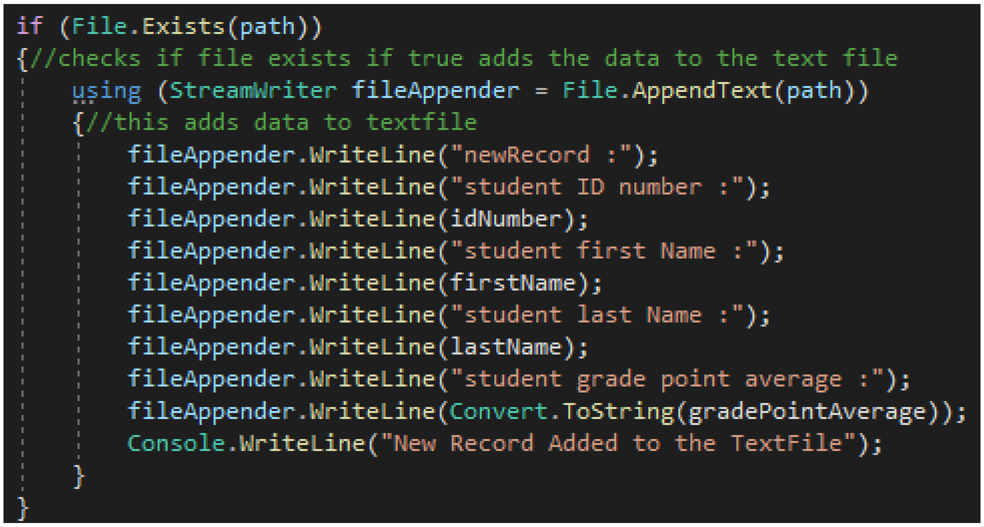
```
   catch (IOException ex)
   {//checks for io errors
       Console.WriteLine($"error occured:{ex} in entry point");
       throw ex;//rethrow
   }
```



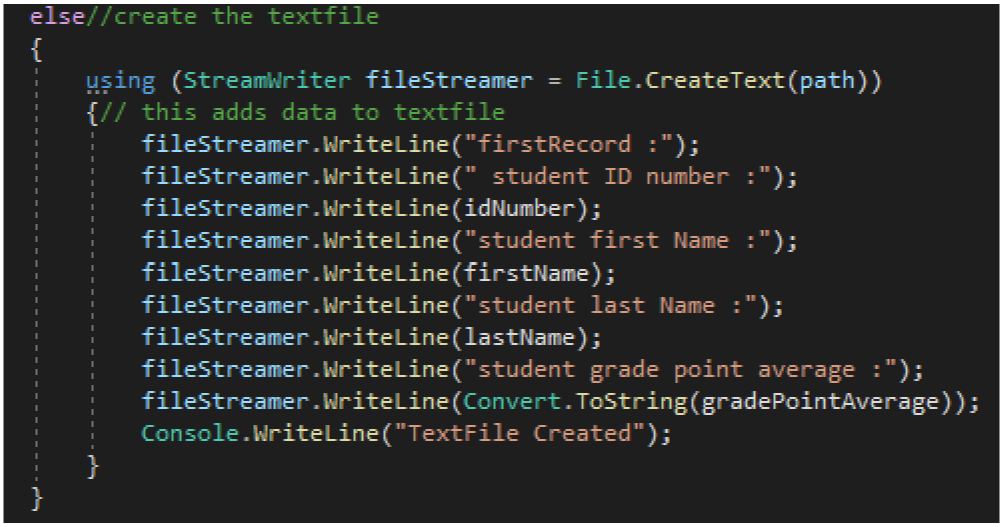
```
    catch (ArithmeticException ex)
    {
        Console.WriteLine($"error occured:{ex} in entry point");
    }
}
```



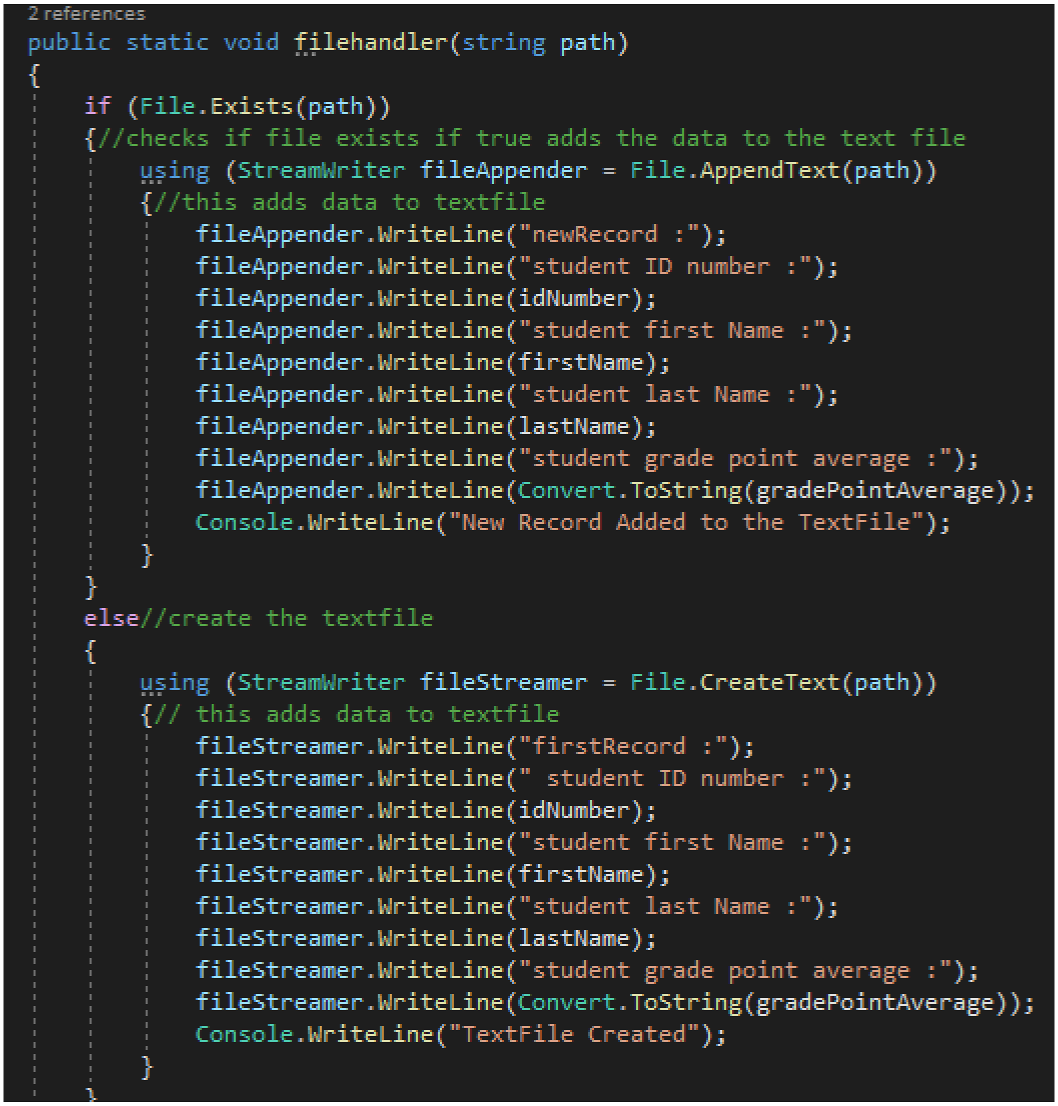
```
    2 references
    public static void filehandler(string path)
    {
```



```
if (File.Exists(path))
{//checks if file exists if true adds the data to the text file
    using (StreamWriter fileAppender = File.AppendText(path))
    {//this adds data to textfile
        fileAppender.WriteLine("newRecord :");
        fileAppender.WriteLine("student ID number :");
        fileAppender.WriteLine(idNumber);
        fileAppender.WriteLine("student first Name :");
        fileAppender.WriteLine(firstName);
        fileAppender.WriteLine("student last Name :");
        fileAppender.WriteLine(lastName);
        fileAppender.WriteLine("student grade point average :");
        fileAppender.WriteLine(Convert.ToString(gradePointAverage));
        Console.WriteLine("New Record Added to the TextFile");
    }
}
```



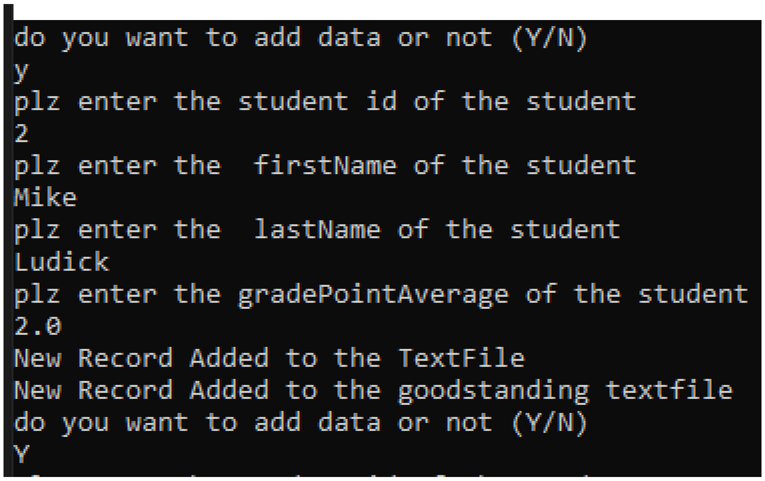
```
else//create the textfile
{
    using (StreamWriter fileStreamer = File.CreateText(path))
    {// this adds data to textfile
        fileStreamer.WriteLine("firstRecord :");
        fileStreamer.WriteLine(" student ID number :");
        fileStreamer.WriteLine(idNumber);
        fileStreamer.WriteLine("student first Name :");
        fileStreamer.WriteLine(firstName);
        fileStreamer.WriteLine("student last Name :");
        fileStreamer.WriteLine(lastName);
        fileStreamer.WriteLine("student grade point average :");
        fileStreamer.WriteLine(Convert.ToString(gradePointAverage));
        Console.WriteLine("TextFile Created");
    }
}
```



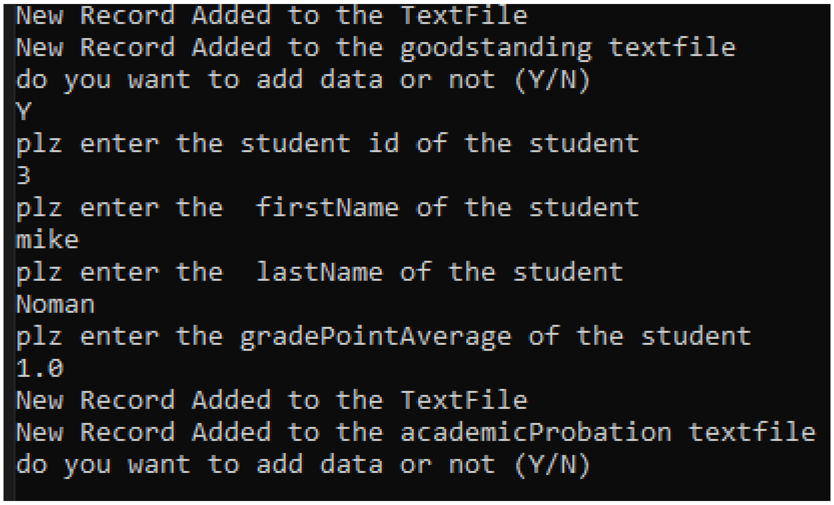
```
public static void filehandler(string path)
{
    if (File.Exists(path))
    {//checks if file exists if true adds the data to the text file
        using (StreamWriter fileAppender = File.AppendText(path))
        {//this adds data to textfile
            fileAppender.WriteLine("newRecord :");
            fileAppender.WriteLine("student ID number :");
            fileAppender.WriteLine(idNumber);
            fileAppender.WriteLine("student first Name :");
            fileAppender.WriteLine(firstName);
            fileAppender.WriteLine("student last Name :");
            fileAppender.WriteLine(lastName);
            fileAppender.WriteLine("student grade point average :");
            fileAppender.WriteLine(Convert.ToString(gradePointAverage));
            Console.WriteLine("New Record Added to the TextFile");
        }
    }
    else//create the textfile
    {
        using (StreamWriter fileStreamer = File.CreateText(path))
        {// this adds data to textfile
            fileStreamer.WriteLine("firstRecord :");
            fileStreamer.WriteLine(" student ID number :");
            fileStreamer.WriteLine(idNumber);
            fileStreamer.WriteLine("student first Name :");
            fileStreamer.WriteLine(firstName);
            fileStreamer.WriteLine("student last Name :");
            fileStreamer.WriteLine(lastName);
            fileStreamer.WriteLine("student grade point average :");
            fileStreamer.WriteLine(Convert.ToString(gradePointAverage));
            Console.WriteLine("TextFile Created");
        }
    }
```

Output of all questions

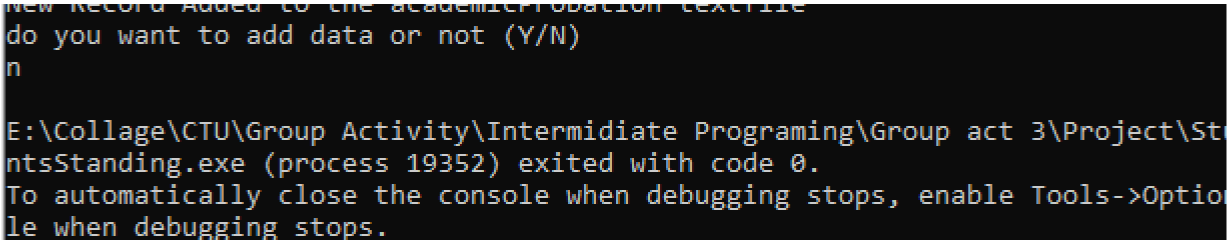
Output of Question 1:



```
do you want to add data or not (Y/N)
y
plz enter the student id of the student
2
plz enter the  firstName of the student
Mike
plz enter the  lastName of the student
Ludick
plz enter the gradePointAverage of the student
2.0
New Record Added to the TextFile
New Record Added to the goodstanding textfile
do you want to add data or not (Y/N)
Y
```

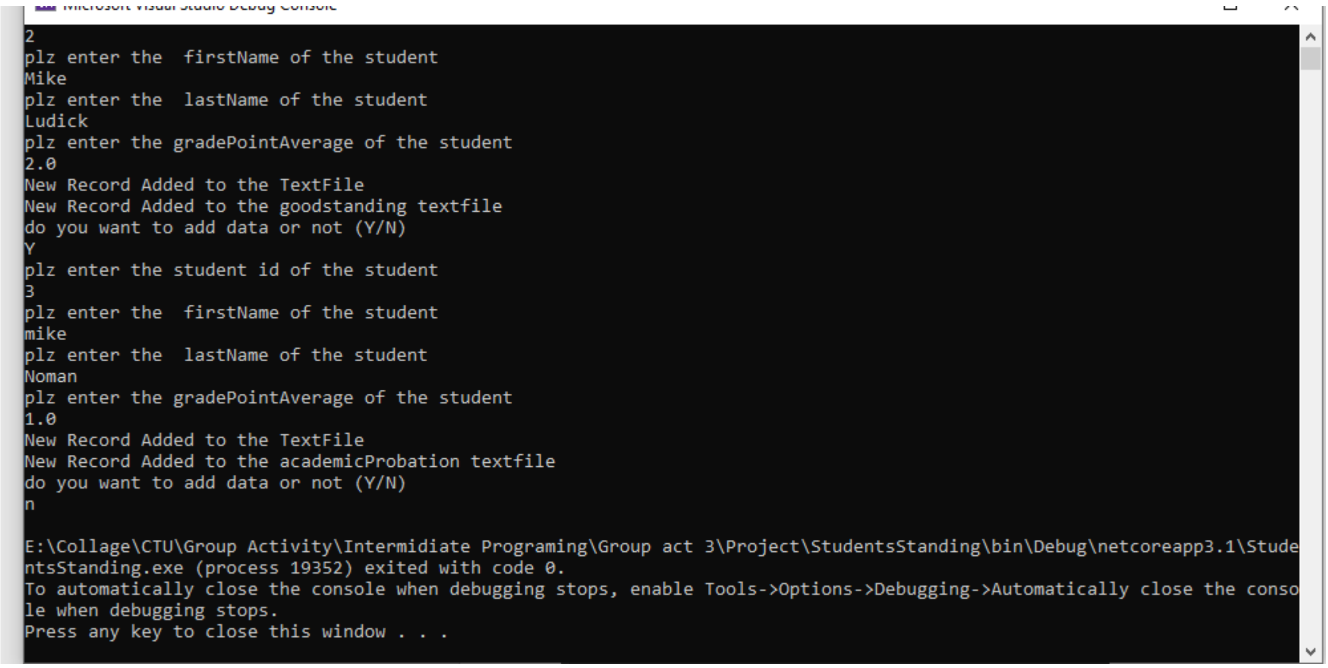


```
New Record Added to the TextFile
New Record Added to the goodstanding textfile
do you want to add data or not (Y/N)
Y
plz enter the student id of the student
3
plz enter the  firstName of the student
mike
plz enter the  lastName of the student
Noman
plz enter the gradePointAverage of the student
1.0
New Record Added to the TextFile
New Record Added to the academicProbation textfile
do you want to add data or not (Y/N)
```



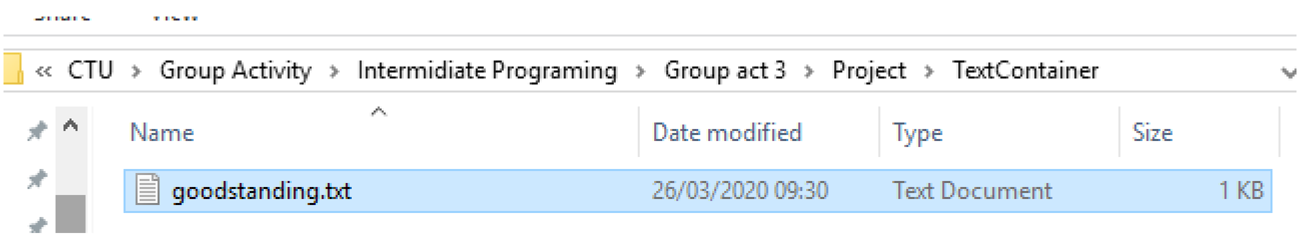
```
New Record Added to the academicProbation textfile
do you want to add data or not (Y/N)
n

E:\Collage\CTU\Group Activity\Intermidiate Programing\Group act 3\Project\St
ntsStanding.exe (process 19352) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Optio
le when debugging stops.
```



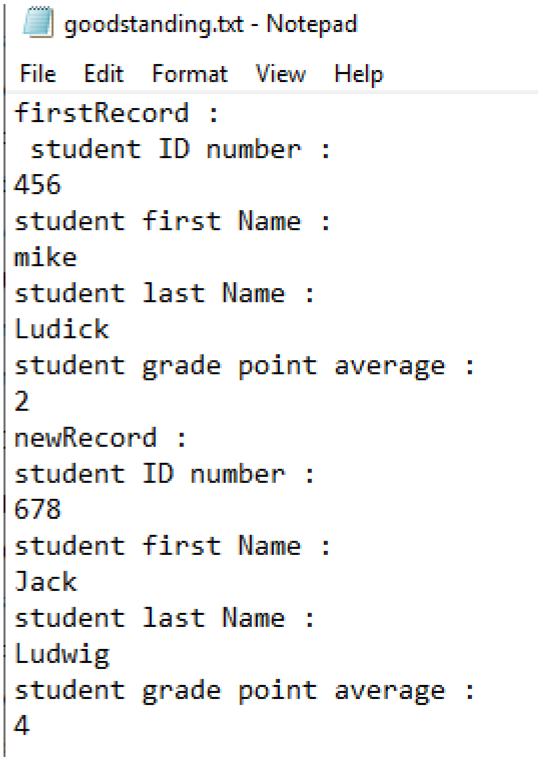
```
2
plz enter the  firstName of the student
Mike
plz enter the  lastName of the student
Ludick
plz enter the gradePointAverage of the student
2.0
New Record Added to the TextFile
New Record Added to the goodstanding textfile
do you want to add data or not (Y/N)
Y
plz enter the student id of the student
3
plz enter the  firstName of the student
mike
plz enter the  lastName of the student
Noman
plz enter the gradePointAverage of the student
1.0
New Record Added to the TextFile
New Record Added to the academicProbation textfile
do you want to add data or not (Y/N)
n

E:\Collage\CTU\Group Activity\Intermidiate Programing\Group act 3\Project\StudentsStanding\bin\Debug\netcoreapp3.1\Stude
ntsStanding.exe (process 19352) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

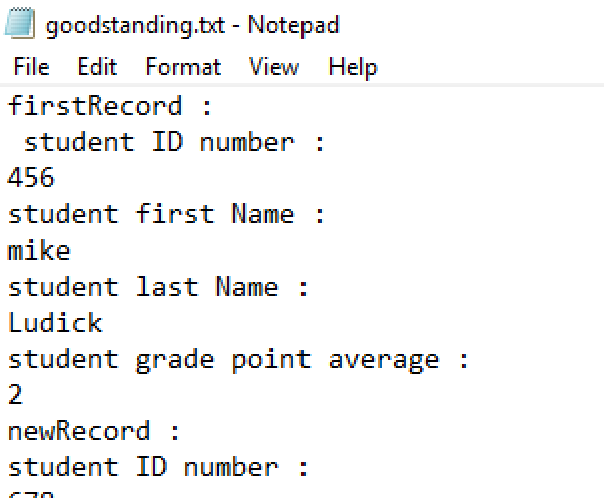


« CTU › Group Activity › Intermidiate Programing › Group act 3 › Project › TextContainer

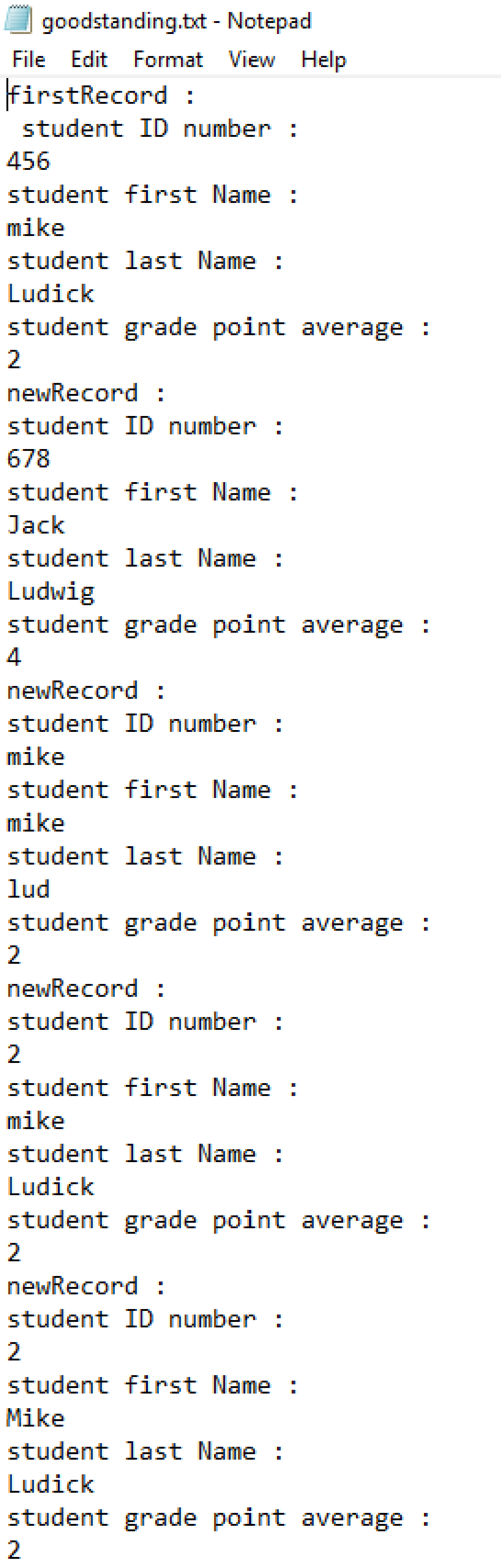
Name	Date modified	Type	Size
goodstanding.txt	26/03/2020 09:30	Text Document	1 KB



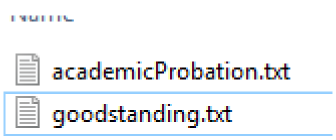
```
firstRecord :
 student ID number :
456
student first Name :
mike
student last Name :
Ludick
student grade point average :
2
newRecord :
student ID number :
678
student first Name :
Jack
student last Name :
Ludwig
student grade point average :
4
```



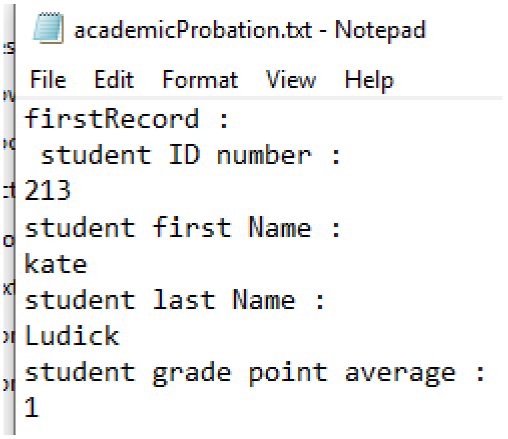
```
firstRecord :
 student ID number :
456
student first Name :
mike
student last Name :
Ludick
student grade point average :
2
newRecord :
student ID number :
678
```



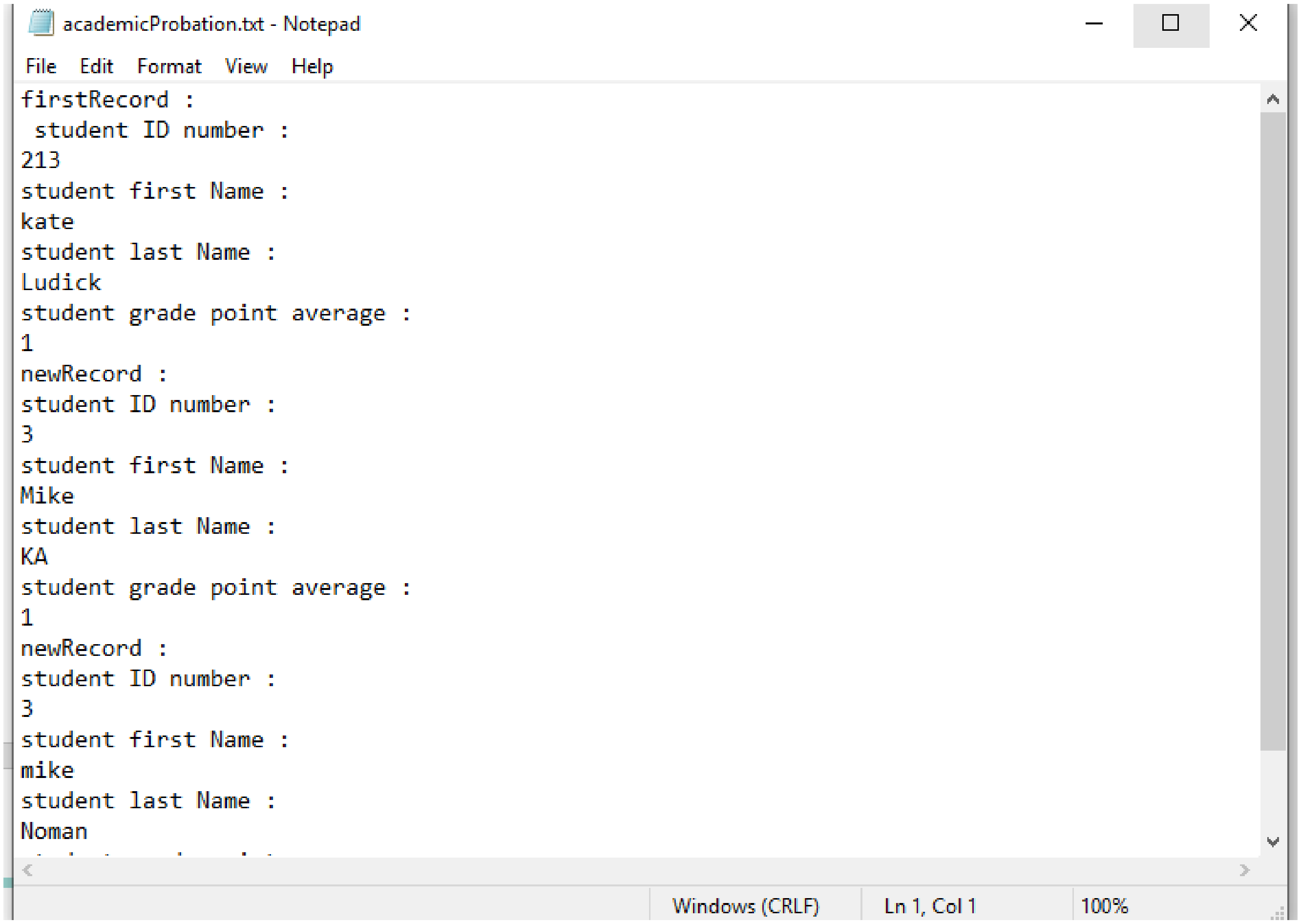
```
firstRecord :
 student ID number :
456
student first Name :
mike
student last Name :
Ludick
student grade point average :
2
newRecord :
student ID number :
678
student first Name :
Jack
student last Name :
Ludwig
student grade point average :
4
newRecord :
student ID number :
mike
student first Name :
mike
student last Name :
lud
student grade point average :
2
newRecord :
student ID number :
2
student first Name :
mike
student last Name :
Ludick
student grade point average :
2
newRecord :
student ID number :
2
student first Name :
Mike
student last Name :
Ludick
student grade point average :
2
```



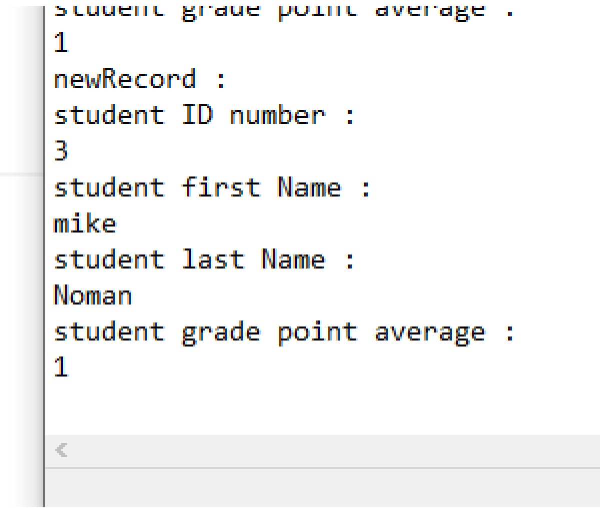
academicProbation.txt
goodstanding.txt



```
firstRecord :
 student ID number :
213
student first Name :
kate
student last Name :
Ludick
student grade point average :
1
```



```
firstRecord :
 student ID number :
213
student first Name :
kate
student last Name :
Ludick
student grade point average :
1
newRecord :
student ID number :
3
student first Name :
Mike
student last Name :
KA
student grade point average :
1
newRecord :
student ID number :
3
student first Name :
mike
student last Name :
Noman
```



```
student grade point average :
1
newRecord :
student ID number :
3
student first Name :
mike
student last Name :
Noman
student grade point average :
1
```

Bibliography & References:

Websites:

GeeksforGeek : https://www.geeksforgeeks.org/
C-Sharp corner: https://www.c-sharpcorner.com/

Books:

Microsoft Visual C# Step by Step, 9th Edition

Appendix

No Extra files or media.

Except for the project itself