Objective

To practice and learn the following coding skills:

- Working with Classes and objects
- Working with text and binary files
- Using arrays and predefined methods such as Split

Instructions

We would like to write a program that reads from a text file the information of students and calculates the GPA of each student and writes the name and GPA of the student into a binary file. We use the concept of the class and object to implement this program.
Step 1: write your full name and student ID as a multiline comment on top of your program.
Step 2: define a Class named Student. This class has 3 properties: Name, Grades, and GPA. Name is a string, holding the name of the student, Grades is an array of floats holding grades of several courses, and GPA is a float type holding the GPA of all grades of the student. This class has one method that is called GetGPA. This method has no input and no return value. This method uses Grades array and calculates the GPA of those grades and store it in the property GPA of the object student.
This class has a constructor. It has 1 input parameter. The input parameter to the constructor is the number of courses that is passed to this constructor when the object is created. The constructor receives the number of courses and creates an array of that size and assigns it to the class property Grades.
For your information, there is one sample code for defining Classes and objects uploaded on Blackboard along with the assignment.
Step 3: in the Main method, define one const for the number of courses. The number of courses must match the number of grades (number values) in the text file.
You can define a variable to work as a counter for displaying on Console, phrases such as "Reading record **1**…" , "Reading record **2**…",… (please run the executable file provided to you along with this assignment to better understand how this program works and how it displays messages on the console.)
Step 4: In your code, write code to open the text file for reading. Also, code for opening a binary file for writing.
Step 5: Create a new object from the class Student. When you create your object using the constructor, you must specify the number of courses, that is:

```
Student student = new Student(NO_OF_COURSES);
```

Step 6: Write a loop that reads the file line by line until reaches the end of text file. This is the pseudo-code for this loop

While not reached to the end of the stream {

- *Read a line from the text file and store it to a string*
- *To inform the user that your program is running, update the console screen with a phrase like "Reading record 1…"*
- *Split this string to items. The space is the separator (hint: use Split method: read here how to use this method: https://docs.microsoft.com/en-us/dotnet/csharp/how-to/parse-strings-using-split)*
- *Store the name of the student to the property Name of the created object*
- *Through a loop, convert grades read from the file, one by one, from string to float and store them in the array Grades of the object.*
- *Call the method GetGPA to calculate the grades that are stored in the Grades array.*
- *Write the name of the student and their GPA to the binary file.*

}

You are provided with an executable file and text input file with some sample data for your test. Make sure your program behaves just like this executable file of the assignment. You can add more data to the text file and test your program. You can see two screenshots of my program with two sets of data.

Requirements

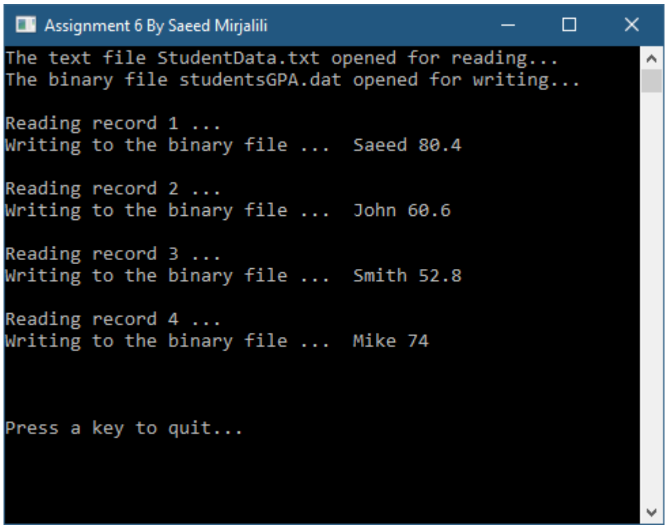
1. On very top of your program, before using directives, update the multiline comment with your name.
2. Make sure your code looks neat and tidy. Press Ctrl+KD in Visual Studio to format your code based on coding styles.
3. Respect Naming Conventions for constants (ALL_CAPS) , variables (camelCase), Method names (PascalCase).

Submissions

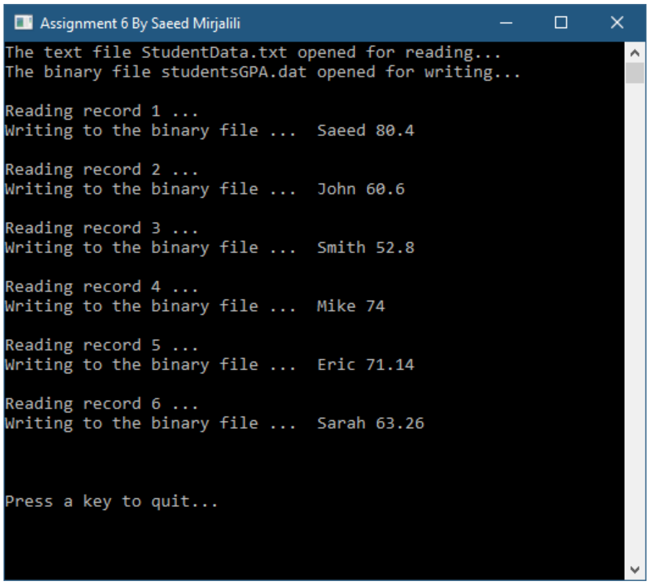
Please submit the following items to the corresponding assignment folder on BlackBoard prior to the deadline:

- A zip/archive folder containing the source code files (.cs files)(two .cs files for this assignment as you create Student class). Rename the zip file name as a#_xy.zip where # is assignment number, x is your first name and y is your last name. For example, when I submit my work for assignment 2, I will rename the zip file as a2_SaeedMirjalili.zip
- If you do not know how to zip a file or folder, search Google for it (for example click here)

Sample Running Screenshots



```
The text file StudentData.txt opened for reading...
The binary file studentsGPA.dat opened for writing...

Reading record 1 ...
Writing to the binary file ...  Saeed 80.4

Reading record 2 ...
Writing to the binary file ...  John 60.6

Reading record 3 ...
Writing to the binary file ...  Smith 52.8

Reading record 4 ...
Writing to the binary file ...  Mike 74



Press a key to quit...
```



```
The text file StudentData.txt opened for reading...
The binary file studentsGPA.dat opened for writing...

Reading record 1 ...
Writing to the binary file ...  Saeed 80.4

Reading record 2 ...
Writing to the binary file ...  John 60.6

Reading record 3 ...
Writing to the binary file ...  Smith 52.8

Reading record 4 ...
Writing to the binary file ...  Mike 74

Reading record 5 ...
Writing to the binary file ...  Eric 71.14

Reading record 6 ...
Writing to the binary file ...  Sarah 63.26



Press a key to quit...
```