**CS 1400 Fundamentals of Programming**

**Programming Project #9**

**A Better Bowling Team Scores Program**

**Version 1.0**

**Objective:**

At the completion of this project, you will have created an application that:

* Gets input from the user and stores it in an array a partially filled,
* Does computations on the array data,
* Sorts the array,
* Uses arithmetic expressions, assignment, and control structures,
* Uses the Split method of the string class to parse user input, and
* Formats output and sends it to the Console or GUI Control(s).



**Project:**

Your friends loved the previous program that you wrote to keep track of their bowling scores, but they would like you to make a few additions and modifications. In particular, they would like the program to display the list of bowling scores, displayed from highest to lowest. Each person's name should be displayed and then their score on the same line. This new improved program should work as follows:

* At the end of each game, the program asks you to record the scores for each team member. You type in their bowlers name and that person's score for the game on a single line.
* Your program uses the Split method to parse the input into a partially filled arrays. The name is then stored in an array of strings and the score is stored in an array of integers.
* When there are no more players to input, just enter an empty line.
* The program will now sort the bowling scores from highest to lowest. As you sort the scores, you must also sort the names array in exactly the same order as the scores array. **Note:** You must write your own sorting algorithm -- do not use any of the built in sorting algorithms in the .net library.
* The bowling scores are now displayed, in order from highest to lowest, as shown in the sample output below. If anyone bowls a perfect game (score of 300), put an asterisk by their name.
* You are to also display the team average.

**Programming Considerations**

As in the previous project, all of the business logic for your bowling team should be encapsulated into a domain class of your own design. Remember this class is responsible to store all the data and perform all of the tasks. ***Do not*** have the user interface do any of the business logic.

Your program should work for any number of players on a team, up to 10.

You may choose to do this assignment as a Console or as a GUI program. The class that you design for your bowling team should work the same in either user interface style of program, i.e. the user interface must be isolated from your business logic class. If you choose to create a GUI program, you will need to display your output in a multi-line format. I chose to use a TextBox with its MultiLine Property set to true. If you do this, you will have to build the Text for the TextBox using code like this:

**string outputLine = "";  
 outputLine += first line of data + Environment.NewLine;  
 outputLine += second line of data + Environment.NewLine;  
 etc.  
 theTextBox.Text = outputline;**

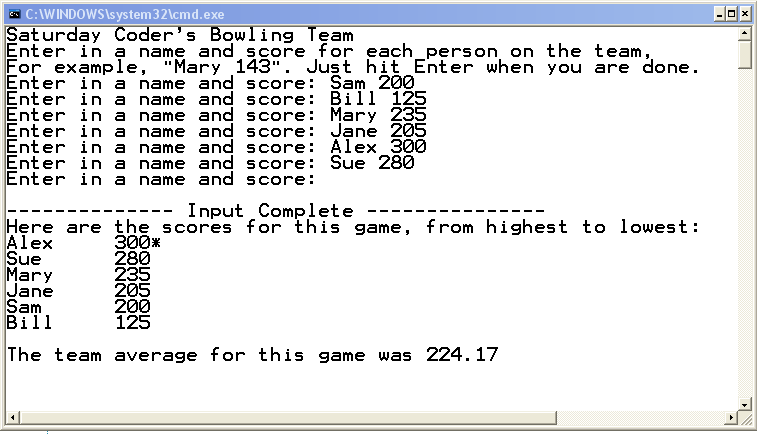
Format and document your code in accordance with the Course Style Guidelines.  Include an appropriate Project Prolog and Method Prologs.  Submit your entire zipped Project folder to Canvas.

**File(s) to Submit:**

Place your entire Project folder into a zip file and name the zip file  
Proj\_09\_your-initials\_V1.0.zip. For example, I would name my file Proj\_09\_DAF\_V1.0.zip. Submit this assignment as Project #9 on Canvas.

|  |  |  |
| --- | --- | --- |
|  | **Grading Checklist** |  |
| # | Program | C(correct)  X(incorrect) |
| 1 | Meets & works to specifications | 6 points |
| 2 | Error Free, elegant & efficient | 4 points |
| 3 | Pseudo-Code | -3 points |
| 4 | Style Guidelines | -2 points |
| 6 | Source Files(s) & Formatting | -2 points |
| 7 | Project Prolog | -1 points |
| 8 | Function Prologs | -1 points |
| 9 | Zip Filename | -1 points |
| 10 | Lab & Project Names | -1 points |
| 11 | Zip File is invalid or will not unzip | Lab = 0 pts |
|  | Total Points | 10 | 0-9 |

**Sample Output for a Console Application:**



You can get a Console executable that runs correctly on Canvas!

You can get a GUI executable that runs correctly on Canvas!