Mr. Bellavia

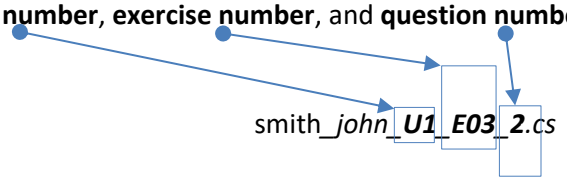
EXERCISE 01 – 1-DIMENSIONAL (1D) ARRAYS

IMPORTANT: Before submission, make a copy of your **'Program.cs'** file for each question and then rename each file to the following:

File Names:

- *last_name_first name*_U4_E01_1_2.cs
- *last_name_first name*_U4_E01_3.cs
- *last_name_first name*_U4_E01_4.cs
- *last_name_first name*_U4_E01_5.cs
- *last_name_first name*_U4_E01_6.cs

Note: Along with last name and first name, make sure the end of the filename (i.e., before the **.cs**) has the **unit number**, **exercise number**, and **question number.** For example:



SUBMIT 1 FILE FOR QUESTIONS 1 & 2:

1. Create and **populate** (to 'populate', add your own data using braces **{ }**) the following arrays:
   - An array to hold all courses for each period of a student timetable (should be a string array with 5 spots).
   - An array to hold all marks for each course (should be a double array with 5 spots)

2. With the above two arrays, write a program that asks the user which period they wish to inquire on (should be an integer from 1 to 5). Use this integer as your index to access the appropriate course & mark from the arrays and output the course and mark to the screen. Your program should have a little bit of error checking (i.e., make sure the index requested is not out of bounds).

---

3. Create and **populate** an array to hold the names of 5 different fruits. Ask the user for the name of a fruit and iterate (traverse) through the array to see if that fruit name exists in the array. (You may want to use .ToLower() or .ToUpper() so that there is no case sensitivity).

4. Create an **empty** array able to hold 5 integers. Populate this array by traversing through this array while asking the user to input an integer for each spot in the array on every iteration. Once completed, traverse the array again to determine the highest and lowest value as well as the average. Finally, output the highest, lowest, and average to the screen. **Note:** You only need to use one traversal for this question, but you can use two if you wish.

5. Create and **populate** two integer arrays 'a' and 'b' containing 5 integers each. Now create a third **empty** integer array 'c' of size 10 and use a **single** for-loop to copy all the integers in 'a' into the first 5 spots of 'c' and all the integers in 'b' into the last 5 spots of 'c'.

Mr. Bellavia

6. **[THINK]** Create and **populate** an array of 5 doubles. Ask the user to input a value 'v' and an index 'i'. Your program should replace the value at index 'i' in the array with the value 'v' while shifting each element to the right and dropping the last element. For example:

Array of 5 doubles:

3.11	8.54	6	3.2	12.98
0	1	2	3	4

User inputs 4.5 for 'v' and 2 for 'i' which results in:

3.11	8.54	4.5	6	3.2
0	1	2	3	4