Project #1 – Mathematical and Array Manipulation Utility

DUE DATE: 01/28/2020 – End of Day

Description:

Your co-worker has implemented the skeleton of an application with all supporting unit tests. However, your co-worker has been re-assigned to a different team and you are now required to implement the application. Your co-worker insists that all unit tests are developed exactly as they should be and the only thing you should need to do is get all of the unit tests to pass. Your objective is to ensure that all defined unit tests pass. No changes to the unit tests or form are permitted. The only two classes that require completion are 'ArrayFunctions.cs' and 'MathFunctions.cs'.

Skills Used: reinforce C# fundamentals, file handling, unit tests, arrays, using existing classes

User Experience:

1. At Startup, the user may utilize the mathematical functions or array manipulation functions
2. If a user selects the mathematical functions, he or she may:
    a. Add two numbers
    b. Subtract two numbers
    c. Multiply two numbers
    d. Divide two numbers
    e. Calculate the modulus of two numbers
    f. Calculate raising a number to a power
    g. Display 'Error' when an invalid calculation is performed
3. If a user selects the array functions, he or she may:
    a. Store the current state of the 5 data values to a persistent file
    b. Load the state of the 5 data values from a persistent file
    c. Randomize the results on the screen
    d. Shift the displayed items to the right with the right-most item becoming the first item
    e. Shift the displayed items to the left with the left-most item becoming the last item
    f. Reverse the order of the items displayed
4. No input validation is performed on the input boxes, so a user could enter any textual entry. This is accounted for in the unit tests present in the project files.

Project Instructions

1. Clone or download the repository for this project to your local machine. If you have troubles retrieving the project from the repository URL, please contact the instructor for assistance.

    GITHUB PROJECT URL: https://github.com/gottjl01/cs203-project1-cs
    GITHUB REPOSITORY URL: https://github.com/gottjl01/cs203-project1-cs.git

2. The main form (frmMain) is already implemented for you and must not be modified.
3. The unit tests are all located in the following project file location:
    Tests\Functions\ArrayFunctionsTests.cs
    Tests\Functions\MathFunctionsTests.cs
4. You **will need** an internet connection to restore the nuget packages in this project.
5. The unit tests **must not** be modified.
6. For full credit on this project, all unit tests must pass and proper documentation added.

Code Documentation

Code comments are meant to provide a brief explanation in areas where the code is less self-explanatory. I ask that you use common sense and critically think about the places a reader of your code might need some guidance. In this project, I have provided many comments in the areas that are pre-implemented that you have been instructed not to modify. However, any method, field or property that you create or modify must have comments added using the standard IDE behavior in the Visual Studio IDE by typing three slashes (///) above a method, property or field. This will cause the IDE to automatically create the commenting structure that you will need to fill in. All parameters and return information must be completed.

Submission

Zip your assignment, including all source, project and solution file(s) and submit the archive through Blackboard by the due date. Late projects will be accepted with a one-week grace period, but no later barring extenuating circumstances, which must be communicated and approved by the professor.

Grading

Your grade is determined by the following rubic:

Scoring Rubic	
Assignment Task	**Points**
Mathematical Functions: Add Unit Tests	3
Mathematical Functions: Subtract Unit Tests	3
Mathematical Functions: Multiply Unit Tests	3
Mathematical Functions: Divide Unit Tests	4
Mathematical Functions: Modulus Unit Tests	4
Mathematical Functions: Power Unit Tests	5
Array Functions: Constructors Unit Tests	6
Array Functions: Store Unit Tests	4
Array Functions: Load Unit Tests	3
Array Functions: Randomize Unit Tests	3
Array Functions: Shift Left Unit Tests	2
Array Functions: Shift Right Unit Tests	2
Array Functions: Reverse Unit Tests	2
Class, Property and Method Documentation	6
TOTAL	**50**

Extra Credit

You may earn additional credit on this project by implementing the same project using VB.NET. You can find the online github repos of the VB.NET version of these projects at the following URLs:

GITHUB PROJECT URL: https://github.com/gottjl01/cs203-project1-vb
GITHUB REPOSITORY URL: https://github.com/gottjl01/cs203-project1-vb.git

Please include the VB.NET solution with your normal project solution. It is best to place the two solutions in different folders and ZIP them up together for the Blackboard submission.

NOTE: The purpose of the extra credit is to encourage you to learn how to convert C# code into VB.NET code using the .NET Framework. DO NOT utilize any online or downloaded tools to automate this process. If your project is detected as having used such tools, you will receive zero extra credit, lose extra credit earned on any other projects and become ineligible for future extra credit on subsequent projects.