Methods Assignment Rubric

Scalable Data Infrastructures: MDV2330

Bare Minimum Requirements

These requirements must be satisfied before any points are awarded. Failing to meet these requirements will result in a zero (0) grade.

1. Working C# file with no major syntax errors and no runtime errors.
2. You must submit the whole project folder and not just the .cs file.

Topic	*%*	*Excellent (100%)*	*Good (75%)*	*Fair (30%)*	*Poor (0%)*
Technical					
Naming	5	The submitted files follow the correct naming convention of LastName_FirstName_Methods.			Files are not named properly.
Programming Fundamentals					
Painting A Wall	15	A custom method is created and works correctly.	There are minor errors in the function but the output is correct.	Custom function is not present, not meaningful, or does not work properly	
Stung!	15	A custom method is created and works correctly.	There are minor errors in the function but the output is correct.	Custom function is not present, not meaningful, or does not work properly	
Reverse It	15	A custom method is created and works correctly.	There are minor errors in the function but the output is correct.	Custom function is not present, not meaningful, or does not work properly	
Arguments & Parameters	20	Arguments & Parameters are present and working in every function.	Arguments & Parameters are present in every function but need minor tweaks to work correctly.	Arguments and parameters are missing and/or there are major erorrs.	Arguments & parameters are not in the code or are not used correctly.
Return Value	20	Return values are present and working in each of the functions.	Return values are present in each of the functions but need minor tweaks.	Return values are missing and/or there are major erorrs.	No functions use return values or they are not working.
Test Values	10	Test values are present in a comment at the end of code and are correct.			No test values are present

Activity: Methods Assignment

OVERVIEW:

For this assignment, these exercises are designed to contribute to your experience with functions, arguments, parameters and returning values

LEVEL OF EFFORT:

This activity should take approximately 240 to complete. It will require:

- 0m Research
- 15m Prep & Delivery
- 225m Work

If you find that this activity takes you significantly less or more time than this estimate, please contact me for guidance.

READING & RESOURCES:

Methods - Rubric (necessary)
The rubric on the first page of this document outlines the points for the assignment.
Make sure you check off each one as done before submitting your assignment!

OBJECTIVES:

Successful completion of this activity will show that you can do the following:

- Identify reusability of code within functions.
- Use arguments and parameters and returning to pass values and variables between functions.
- Organize code into reusable blocks with functions.
- Constructing parameters with appropriate data types.

INSTRUCTIONS:

1. Before you begin, you should read the rubric on page 1. This is extremely important, as it will tell you exactly how this assignment will be graded.

2. Create a project called **Lastname_Firstname_Methods**.

3. In this assignment you will be given (3) different problems to answer. For each one you must do the following:
    a. Label the section of code appropriately.
    b. Create a custom method for each of the (3) problems inside of (1) project.
    c. Prompt the user for the required input in the main method.
    d. Validate and convert each user prompt to insure that the user is typing in a valid response.
    e. Send in the user's input as arguments to the custom method.
    f. The custom method should perform the task and save the result in a variable.
    g. The result variable must be returned to the main method and caught by a variable.
    h. Print out the result of each function to the console in the main method.
    i. Use only code and techniques learned in this class.

4. Place your name, date, and assignment at the top of your code in a multi-lined comment.
5. Make sure to comment every important line of code so that you are explaining exactly what you are trying to do.
6. Your code should give the user meaningful output. So, after your calculations are complete, your code should report back to the user the final values with a Console.WriteLine().
    a. This should contain the variables that you calculated and a concatenation text string that describes the value.
    b. e.g. Console.WriteLine ("The area of the rectangle is " +calcArea+ "!");
7. Zip your whole project folder and upload this file to FSO.

TURNING IT IN:

- Double-check that you’ve commented your code (You can’t comment too much).
- Compress your **Lastname_Firstname_Methods** folder into one zipped file. It should be named **Lastname_Firstname_Methods.zip**
- Upload this zipped file to FSO. This is the file I will unzip and run to verify it works and review your code.
- You must zip the whole folder and not just the one individual C# file. If you only submit a .cs file you will get a **zero** for the whole project.

Don’t Forget:

Make sure your project follows this list of criteria:

- The result should appear in the console and include an explanation of the result.
  - **Good example of console print out:** The volume of the sphere is 26 feet cubed.
  - **Bad example of console print out:** 26
- Final output should use string concatenation.
- Comment every line of code (describe what each line is doing in English). Do NOT just label sections of your code.