Loops 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_LogicLoops.			Files are not named properly.
Programming Fundamentals					
Prompts	10	The user is prompted for all of the "User Inputs" with WriteLine and information is stored in a variable.	There are minor errors in prompting the user.	There are major errors in prompting but it is at least attmepted.	No prompts are present. (Zero for the entire project.)
Validation	15	All user prompted variables are correctly validated using a while loop and the correct conversion and/or test.	User variables are validated/converted but there are minor errors	User variables are validated/converted but there are major errors	User prompted variables are not validated.
Piggy Bank	20	Problem is soved correctly using a for loop. The loop should do the addition for each month and output the new total.	There are minor errors in this problem or there is no array in the problem.	There are major errors in the for loop.	The for loop is missing and/or is completely wrong.
3, 2, 1 Blast Off	20	Problem is soved correctly using a for loop. The loop should do the subtraction and output the countdown for the user.	There are minor errors in this problem.	There are major errors in the for loop.	The for loop is missing and/or is completely wrong.
Darn Good Donuts	20	While loop is setup correctly and works correctly to solve the problem.	There are minor errors in the for loop.	There are major errors in the while loop.	The while loop is missing and/or is completely wrong.
Test Values	10	Test values are present in a comment at the end of code and are correct.			No test values are present

Activity: Logic & Loops

OVERVIEW:

For this assignment, you will be solving given problems using conditionals, relational operators, logical operators and loops.

READING & RESOURCES:

Logic & Loops - 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:

- Determine when to use if, else, and else if conditional statements.
- Formulate conditional statements for dynamic decision-making.
- Employ relational operators to weigh conditions as true or false.
- Utilize logical operators in order to combine multiple conditional statements.
- Create logic to resolve a single solution from a selection of possible solutions.
- Recall and distinguish the structure for while, for and for each loops.
- Formulate loops to repeat blocks of code.
- Recognize each type of loop and when to use them.
- Employ repetition to solve complex problems.
- Identify and use the components of a for loop.

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_LogicLoops.**

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. Prompt the user for each variable that is in the "User Input" Section of that problem.
   c. Validate each user prompt with a **while loop** to insure that the user is typing in a valid response. Remember we check text string different than numbers!
   d. Convert each user response to the **correct** data type, if needed.
   e. The result should be calculated using variables, not literal values when possible.
   f. Create code that will make decisions based on the value of the user's responses.
   g. Print the result of the decision-making to the console using the format given in the "Results" section of that problem.
   h. After each section put in a multi-lined code for the Test Values.
4. Use only code and techniques learned in this class.
5. Place your name, date, and assignment at the top of your code in a multi-lined comment.
6. Make sure to comment every important line of code so that you are explaining exactly what you are trying to do.
7. 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+ "!");
8. 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_LogicLoops** folder into one zipped file. It should be named **Lastname_Firstname_LogicLoops.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.