MadLibs 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_MadLibs.			Files are not named properly.
Programming Fundamentals					
Word Prompts	20	The user is prompted for at least 4 different types of words by using the WriteLine and information is stored in variables.	There are minor errors in prompting the user or you are missing 1 prompt.	There are major errors in prompting but it is at least attempted or your are missing 2 prompts.	No word prompts are present.
Number Prompts	20	The user is prompted for at least 3 numbers by using the Write Line and information is stored inside of variables.	There are minor errors in prompting the user or you are missing one number prompt.	There are major errors in prompting but it is at least attempted or you are missing 2 number prompts.	No number prompts are present.
Conversion	20	The user prompted number variables must be converted into a number data type.			No conversions of data types in code.
Story Output	35	You must write a story that utilizes your user prompted values. This should be output to the console using Console.WriteLines It must contain: 1. Supporting descriptive narrative text. 2. String concatenation. 3. You must use the converted number values inside of your story	Missing one of the previous list.	Missing two.	Output not present or not meaningful.

Activity: MadLibs Style Story Creator

OVERVIEW:

For this assignment, you will be building a MadLibs styled story creator. If you are not familiar with what this is, you ask the user for a series of words and numbers and do not explain what they are for, then they are put inside of a story (hopefully with funny outcomes). You will be prompting the user for words and numbers and writing your own mini –story to fill them into.

READING & RESOURCES:

MadLibs - 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!

MadLibs – This website will show you what MadLibs are and give you a few examples.

Online Version Of MadLibs – This website will allow you to play a game of MadLibs, so you can get a feel for what you are creating yourself.

OBJECTIVES:

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

- Successfully prompt the user and utilize their responses in calculations.
- Convert user responses into usable number data types.
- Create, implement and traverse an array.
- Utilize string concatenation to create a final story output to the Console.

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_**MadLibs.

3. In this assignment you will have the following objectives:

   a. Prompt the user for at least **4** different types of strings and store their responses inside of variables that you create.
      i. Ask for 1 animal
      ii. Ask for 1 name
      iii. Ask for 1 adjective
      iv. Ask for 1 food item
   b. Prompt the user for at least **3** numbers and store their responses inside of variables that you create.
      i. Ask for 1 year
      ii. Ask for 1 cost
      iii. Ask for 1 random number
   c. Convert the user prompted number variables from strings into a number data type and store these inside of new variables that you create.
   d. Brainstorm a new idea of a short story that will utilize all of the prompts that you have asked the user for.
   e. Output your final story to the user using Console.WriteLine(). Make sure it has the following:
      i. Supporting descriptive narrative text.
      ii. String concatenation of all of your user prompted variables.
      iii. Your story must make cohesive sense. Meaning if you ask the user for a name, you must use it as a name.
      iv. You must use the converted number data type variables instead of the string version inside of your story.

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_ MadLibs** folder into one zipped file. It should be named **Lastname_Firstname_ MadLibs.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.