01.04 Documenting & Debugging

Name:

Directions

This assessment has three parts. Choose one of the options provided below:

Scenario Option 1: You're volunteering for an organization whose mission is to raise awareness in high schools about the impact of voting. They would like you to build them a program they can use at events. The program should have a conversation with the user that asks them their current age, why they think voting is important, and tell them how many years until they can register to vote.

Scenario Option 2: Do you enjoy being creative and making up your own scenarios? If you do, then this option is for you! Come up with any scenario for this assignment as long as you do the following:

- Scenario must be about making the world a better place.
- Ensure people reading your code understand your scenario (do this with lots of comments).
- Follow the requirements described below.

Part One: Algorithmic Design and Pseudocode

Use the algorithmic design stage of computational thinking to design a program. Remember that algorithmic design is where you create step-by-step instructions to solve a problem. This must begin with pseudocode, then be translated to Python using your pseudocode as a guide.

Write your pseudocode where indicated below. Your pseudocode should be written in English (not Python), indicate a conversation about improving the world, and must:

be written in a series of steps that reflect algorithmic design

include at least three interactive prompts

assign user input to variables to be used throughout the program. (For example, this could be their name or age.)

print a tip on the importance of your selected topic

Example of expected output: The output for your program should resemble the following screenshot. Your specific results will vary depending on the choices you make about the conversation and the input provided.

Write your pseudocode here:	

Part Two: Algorithmic Design and Program Code

Use the Python IDLE built into this course to code and run your program. Your code must:

Use comments for internal documentation (including a heading with your name, today’s date, and a short description of the program). Remember that you can copy/paste much of your pseudocode for this internal documentation.

use concatenation to join string literals and string values–Hint: `print("This is a string literal " + name + ", that was followed by a value and another string literal.")`

Follow the Python style conventions regarding indentation and the use of white space in your program.

Run successfully and produce output similar to the example above.

When you’ve completed writing your program code, save your work by selecting ‘Save’ in the Python IDLE. When you submit your assignment, you will turn in this Python file separately.

Part Three: Generalize & Assess with a Post Mortem Review

Complete the Post Mortem Review (PMR). Write thoughtful two- to three-sentence responses to all the questions in the PMR chart.

Post Mortem Review Question	**Response**
What was the purpose of your program?	
How could your program be useful in the real world?	
What is a problem you ran into, and how did you fix it?	
Describe one thing you would do differently the next time you write a program.	

How could your program be generalized and useful in other areas?	