Project plan for Project 3a

Start Assignment

Due Sunday by 11:59pm **Points** 20 **Submitting** a file upload **File Types** pdf
Available until Jan 23 at 11:59pm

Introduction

You will create a testing plan and design for project 3a (https://canvas.oregonstate.edu/courses/1946267/assignments/9464112) and submit it as a pdf here in Canvas. You'll submit your code for project 3a on GradeScope as usual (together with the code for projects 3b and 3c). When doing your coding, you're allowed to deviate from the plan you made.

What you must do

Your design document should contain the two following sections:

- **Testing Plan:** Design some tests you can perform to verify that your program meets the given specifications. They should be representative enough that if your program passes them, you would have a high level of confidence that your program is ready to be submitted. Having just a couple of tests will not meet that criterion. Create a table with two columns: one for the description of each test and one for the expected results of each test. These are not meant to be general descriptions, but rather specific concrete examples of inputs and outputs. Try to come up with tests that will check different aspects of the program's functioning, and be creative in thinking of situations that might "break" the program. The tests should check whether your code works correctly when the user enters valid inputs. You don't need to test for wrong inputs.
- **Design:** Describe or draw out your design for how the program should behave using pseudocode or flowcharts, following the "Pseudocode and flowchart guidelines". You do not need to do both pseudocode and a flowchart - just one or the other.

What to turn in

This part must be submitted as a pdf here in Canvas.

Example Testing Plan

Below is an example testing plan for a program that is supposed to determine whether the sum of the integers entered by the user is even or odd. (For examples of what a pseudocode or flowchart design should look like, see the "Pseudocode and flowchart guidelines".)

Example Testing Plan

Input	Output
User enters: 1, 9, 4	Output “even”
User enters: 2, 2, 1, 3, 7	Output “odd”
User enters: 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1	Output “odd”
User enters: 7, -9, 2, 100000	Output “even”
User enters: 187	Output “odd”
User enters: -9000	Output “even”
User enters: 0	Output “even”

Testing Plan and Design

Criteria	Ratings			Pts
coverage of proposed tests	**5 to >4.0 pts** **proposed tests are adequate for making sure the code meets the specifications**	**4 to >0.0 pts** **proposed tests omit cases that should be checked to make sure the code meets the specifications**	**0 pts** **proposed tests are absent or inadequate**	5 pts
correctness of expected results	**5 pts** **expected results are correct for proposed tests**	**3 pts** **there are flaws in the expected results**	**0 pts** **expected results are absent or inadequate**	5 pts
design adheres to pseudocode & flowchart guidelines	**5 pts** **pseudocode is specific without looking like Python; flowchart notation is used correctly**	**3 pts** **pseudocode is vague or looks similar to Python; incorrect use of flowchart notation**	**0 pts** **pseudocode is quite vague or is essentially Python; flowchart notation is largely incorrect**	5 pts
correctness of design logic	**5 pts** **design logic seems sound**	**3 pts** **design logic is somewhat adequate, but has gaps or flaws**	**0 pts** **design logic has many gaps or flaws**	5 pts
				Total Points: 20