Python Exercises 2

Start Assignment

- Due Monday by 11:59pm
- Points 15
- Submitting a file upload
- File Types zip

Python Exercise - 2

NOTE: All coding problems must be done via code - no hard coding of any type is allowed for credit. Any crash or execution errors that are not gracefully handled will result in a mandatory 20% reduction.

Tasks:

Exercise 1: Periodic Payment Calculation (GUI Application)

1. **Objective:**

   - Build a Python GUI application using libraries like **tkinter** that calculates the present value annuity (Pvann) using the formula:

$$\text{Pvann} = \text{PMT} \times \frac{1-(\frac{1}{(1+r)^n})}{r}$$

   where

   - PMT: Periodic Payment
   - r: Interest Rate (in decimal form)
   - n: Number of Payments

2. **Requirements:**

   - Create a GUI form with input fields for:
     - PMT (default: 10000)
     - r (default: 8%)
     - n (default: 20)
   - Provide buttons for Calculate that Calculates Pvann and displays the result on the form and for Reset that Resets all inputs to their default values.
   - Display results dynamically on the form after clicking Calculate.

- Allow users to modify inputs before recalculation.

3. **Sample Result:**

   - For PMT=10000, r=8%, and n=20, the output should be 98181.47.

Exercise 2: Student Class Standing

1. **Objective:**

   - Build a Python application that determines the class standing of a student based on credit hours earned.

2. **Requirements:**

   - Create a Python class named Student with:
     - Private attributes: first_name, last_name, credits.
     - Methods that Adds credit hours to the student, Determines and displays the class standing based on credit hours:
       - Freshman: 24 or fewer credits
       - Sophomore: 25 to 55 credits
       - Junior: 56 to 86 credits
       - Senior: 87 or more credits
   - Build a driver program to:
     - Instantiate four or more Student objects with predefined names and distinct credits ranges.
     - Display their class standings.
   - Class standings should appear in a tabular format in the terminal.

Submission Details:

.zip folder comprising:-

- your 2 .py file. (Exercise1.py and Exercise2.py).
- 2 Screenshots (GUI form output for Exercise 1 and Terminal output displaying class standings for Exercise 2).

Python Exercise 2 Rubric

Criteria	Ratings		Pts
GUI Design and Functionality	**2 pts** **Full Marks** GUI includes fields for PMT, interest rate, and number of payments. Buttons for Calculate and Reset are functional.	**0 pts** **No attempt**	2 pts
Input Validation	**2 pts** **Full Marks** Validates and handles invalid inputs (e.g., non-numeric values) gracefully without crashing.	**0 pts** **No Marks**	2 pts
Formula Implementation	**3 pts** **Full Marks** Formula is correctly implemented, and results match expected output.	**0 pts** **No Marks**	3 pts
User Experience	**1 pts** **Full Marks** Results are displayed dynamically, and reset functionality works as expected.	**0 pts** **No Marks**	1 pts
Class implementation	**3 pts** **Full Marks** Class is fully implemented with private attributes and methods that work as expected.	**0 pts** **No Marks**	3 pts
Driver Program	**3 pts** **Full Marks** Instantiates Student objects and displays their class standings in a tabular format.	**0 pts** **No Marks**	3 pts
Standing Calculation Logic	**1 pts** **Full Marks** Correctly determines the class standing based on credit hour ranges	**0 pts** **No Marks**	1 pts

Total Points: 15