Python Exercise 7

N-sided Polygon class definition and main function (30 pts)

This exercise will help you get familiar with creating classes and objects of your own.

A regular polygon is an n-sided polygon whose sides are of the same length and whose angles all have the same degree. The formula for computing the area and perimeter of a regular polygon is

$$Area = \frac{n \times s^2}{4 \times \tan\left(\frac{\pi}{n}\right)}$$

$$Perimeter = n \times s$$

where n is the number of sides and s is the length of each side.

You will create a class definition and a main function that references information about the regular polygon.

For the class definition:

- Create a class that contains data and methods for a regular polygon.
- Create two variables as hidden data members of the class. One member should store the number of sides (as an int), while the second member should store the length of each side (as a float).
- Create an initializer method. This method only requires the self parameter. (You can initialize the data members to 0 and 0.0 respectively.)
- Create accessor and mutator methods for each of the hidden data. Use the mutators to update the hidden data members.
- Create a class method that accepts no parameters but returns the perimeter of the polygon. It should use the hidden data members of the class to calculate the perimeter.
- Create another class method that accepts no parameters but returns the area of the polygon. It should use the hidden data members of the class to calculate the area.
    - Import the math module to use the PI constant and tan function.

For the main function:

- Create a polygon object using the class you created.
- Create local variables to obtain the number of sides and the length of each side. Prompt the user to enter values for the number of sides and the length of each side. Use input validation to ensure that the number of sides is >=3 and the length of each side is >= 0.1.

- Use mutator methods to update the values in the object and accessor methods to display the current dimensions of the object. (You can reference accessor methods within your print functions).
- Use the class methods to display the polygon's perimeter and area. Format your result to 3 decimal places.

Here is a sample of the output:

```
Enter the number of sides (>=3): 1
Invalid entry. Re-enter the number of sides (>=3): 5
Enter the length of each sides (>= 0.1): 0.00004
Invalid entry. Re-enter the length of each sides (>= 0.1): 3.5
The polygon has 5 sides. Each side is 3.5 units in length.
The perimeter of the polygon is 17.500 units and its area is 21.076 square units.
```

Note:

- **Do not** create an str method for this program. Use your accessor methods instead. In the same, the accessor methods are used to display the updated values of the polygon (after the mutators were used.)

Retail items (30pts)

Write a class called Retail_Item that has the following data and methods:

The data (attributes) for the Retail_Item Class is as follows:

- type of item (e.g. Jacket, shirt)
- amount of item in inventory
- price of item

Define your data so that they are hidden from code outside of the class definition.

Create two methods for this class:

- an __init__method that accepts three parameters (corresponding to the data) in addition to the required parameter.
- a __str__ method that formats the data (in preparation for output).
    - Create formatting strings with field widths and justifications in this part.

Next <u>write a main function</u> that does the following:

- Prompts the user to enter the information for two items
    - You are welcome to create more, but two items are the minimum
- Creates an object to represent each item
    - Note: No loop or list is required for this question
- Displays the information of the two items in a neat table
    - The table header should be created in main so that it aligns with the formatting for the object's data. (Remember, the __str__ method should be used here.)

The main function may be located in the same file as the class definition.

Sample output is shown below:

```
Name of item 1: Jeans
Amount of item 1: 50
Price of item 1: 59.95

Name of item 2: T-Shirt
Amount of item 2: 65
Price of item 2: 19.99

Here is a summary of the items you added:
Item             Amount        Price
--------------------------------------------------
Jeans                50       $59.95
T-Shirt              65       $19.99
```