Exercise: Shapes in Python

Objective:

The objective of this exercise is to practice working with Python classes, class attributes, class methods, encapsulation, properties, inheritance, and overriding methods. You will create a simple system to model different shapes using classes in Python.

Instructions:

1. **Base Class - Shape:**
    - Create a base class named **Shape**.
    - The class should have a constructor that takes a **color** attribute.
    - Implement an **area** method in the base class. This method should be a placeholder and will be overridden by the subclasses.
    - Implement a **display_info** method that prints basic information about the shape.
2. **Subclass - Circle:**
    - Create a subclass named **Circle** that inherits from the **Shape** class.
    - The **Circle** class should have a constructor that takes **color** and **radius** attributes.
    - Override the **area** method to calculate and return the area of a circle using the formula: **area = π * radius^2**, where π is a constant (you can use 3.14).
    - Override the **display_info** method to display information about the circle, including its color, radius, and area.
3. **Subclass - Rectangle:**
    - Create another subclass named **Rectangle** that also inherits from the **Shape** class.
    - The **Rectangle** class should have a constructor that takes **color**, **length**, and **width** attributes.
    - Override the **area** method to calculate and return the area of a rectangle using the formula: **area = length * width**.
    - Override the **display_info** method to display information about the rectangle, including its color, length, width, and area.
4. **Testing:**
    - Instantiate objects of both the **Circle** and **Rectangle** classes.
    - Set appropriate values for the attributes of each object.
    - Call the **display_info** method for each object to ensure that the correct information is displayed.

5. **Additional Challenge:**
    - Create a new shape class with additional attributes and methods.
    - Create a new subclass that inherits from both the **Shape** class and the new shape class.