OOP Rectangle

11/4/2022

90/100 Points

Attempt 1

Review Feedback
11/4/2022

Attempt 1 Score:
90/100

View Feedback

Anonymous Grading: **no**

Unlimited Attempts Allowed

12/9/2022

Details

Preparation

If you are in the DE section please view videos up to Object Oriented Programming

Overview

The purpose of this lab is to give you practice creating your own object. You will be given a main function that expects an instance of the (yet undefined) Rectangle object.

Your job is to determine what attributes and methods the Rectangle class requires and create a class that will satisfy the requirements.

The main function

You can begin with this main function:

```
def main():
    print ("Rectangle a:")
    a = Rectangle(5, 7)
    print ("area:      {}".format(a.area))
    print ("perimeter: {}".format(a.perimeter))

    print ("")
    print ("Rectangle b:")
    b = Rectangle()
    b.width = 10
    b.height = 20
    print (b.getStats())
```

You can begin by copying this code into your editor. While you are allowed (and encouraged) to make temporary changes to this code (mainly commenting out features that are not yet implemented) you may not turn in a main function substantially different than this one.

Expected Output

When the Rectangle class has been properly created, the output should look like the following:

```
Rectangle a:
area:      35
perimeter: 24
Rectangle b:
width:     10
height:    20
area:      200
perimeter: 60
```

Tips

- Add only one feature at a time
- You may need to comment out parts of the main function for testing
- Constructor should take 0, 1, or 2 parameters (illustrating polymorphism)
- Your class should be a subclass of something (illustrating inheritance)
- Your class should have methods and properties (illustrating encapsulation)
- Make your instance variables hidden (using the __ trick)
- Add setter and getter methods for each instance variable
- Use properties to encapsulate instance variable access
- I describe one way to do properties in class. You may discover another approach called a *decorator* online. Either approach is acceptable.
- Not all instance variables are real... Some are derived, and should be read-only
- You may not change the main function (unless you're doing the blackbelt challenge)
- Be sure to add the needed code to run the main function when needed

Submission

Please submit the following on Canvas:

- Your .py file (NOT a link to your pythonanywhere page and NOT a word document)
- A .txt file describing your algorithm (congruent with the requirements for algorithm files described in **the announcement (%24CANVAS_OBJECT_REFERENCE%24/discussion_topics/g6b4bf997b8a8e34ab09edb503b4187b4)** about algorithm files)
- If you are turning in a blackbelt version, submit your blackbelt as a separate .py file from your basic .py file

Blackbelt Challenge

Add a Graphic Interface to the project. Use labels to print out all the output. Additionally, add text fields for height and width, and calculate the area and perimeter using your class.

	File Name	Size	
🗎	OOPRectangle.py.txt	733 Bytes	✓
🗎	OOPRectangle.txt	1.33 KB	✓