Scientific Computing wit… Scientific Computing with Python Projects (/learn/scientific-computing-with-python

Polygon Area Calculator

You will be working on this project with our Replit starter code (https://replit.com/github/freeCodeCamp/boilerplate-polygon-area-calculator).

In this project you will use object oriented programming to create a Rectangle class and a Square class. The Square class should be a subc of Rectangle and inherit methods and attributes.

Rectangle class

When a Rectangle object is created, it should be initialized with `width` and `height` attributes. The class should also contain the followir methods:

- `set_width`
- `set_height`
- `get_area` : Returns area (`width * height`)
- `get_perimeter` : Returns perimeter (`2 * width + 2 * height`)
- `get_diagonal` : Returns diagonal (`(width ** 2 + height ** 2) ** .5`)
- `get_picture` : Returns a string that represents the shape using lines of "*". The number of lines should be equal to the height and th number of "*" in each line should be equal to the width. There should be a new line (`\n`) at the end of each line. If the width or height larger than 50, this should return the string: "Too big for picture.".
- `get_amount_inside` : Takes another shape (square or rectangle) as an argument. Returns the number of times the passed in shape could fit inside the shape (with no rotations). For instance, a rectangle with a width of 4 and a height of 8 could fit in two squares witl sides of 4.

Additionally, if an instance of a Rectangle is represented as a string, it should look like: `Rectangle(width=5, height=10)`

Square class

The Square class should be a subclass of Rectangle. When a Square object is created, a single side length is passed in. The `__init__` metl should store the side length in both the `width` and `height` attributes from the Rectangle class.

The Square class should be able to access the Rectangle class methods but should also contain a `set_side` method. If an instance of a Square is represented as a string, it should look like: `Square(side=9)`

Additionally, the `set_width` and `set_height` methods on the Square class should set both the width and height.

Usage example

```
rect = shape_calculator.Rectangle(10, 5)
print(rect.get_area())
rect.set_height(3)
print(rect.get_perimeter())
print(rect)
print(rect.get_picture())

sq = shape_calculator.Square(9)
print(sq.get_area())
sq.set_side(4)
print(sq.get_diagonal())
print(sq)
print(sq.get_picture())

rect.set_height(8)
rect.set_width(16)
print(rect.get_amount_inside(sq))
```

(/learn) ≡ (/fcc6 882a- 4821- a91e- 117a1

That code should return:

```
50
26
Rectangle(width=10, height=3)
**********
**********
**********

81
5.656854249492381
Square(side=4)
****
****
****
****

8
```

The unit tests for this project are in `test_module.py`.

Development

Write your code in `shape_calculator.py`. For development, you can use `main.py` to test your `shape_calculator()` function. Click "run" button and `main.py` will run.

Testing

We imported the tests from `test_module.py` to `main.py` for your convenience. The tests will run automatically whenever you hit the "run" button.

Submitting

Copy your project's URL and submit it to freeCodeCamp.

Solution Link

ex: https://replit.com/@camperbot/hello

I've completed this challenge

Get a Hint (https://forum.freecodecamp.org/t/462363)

Ask for Help