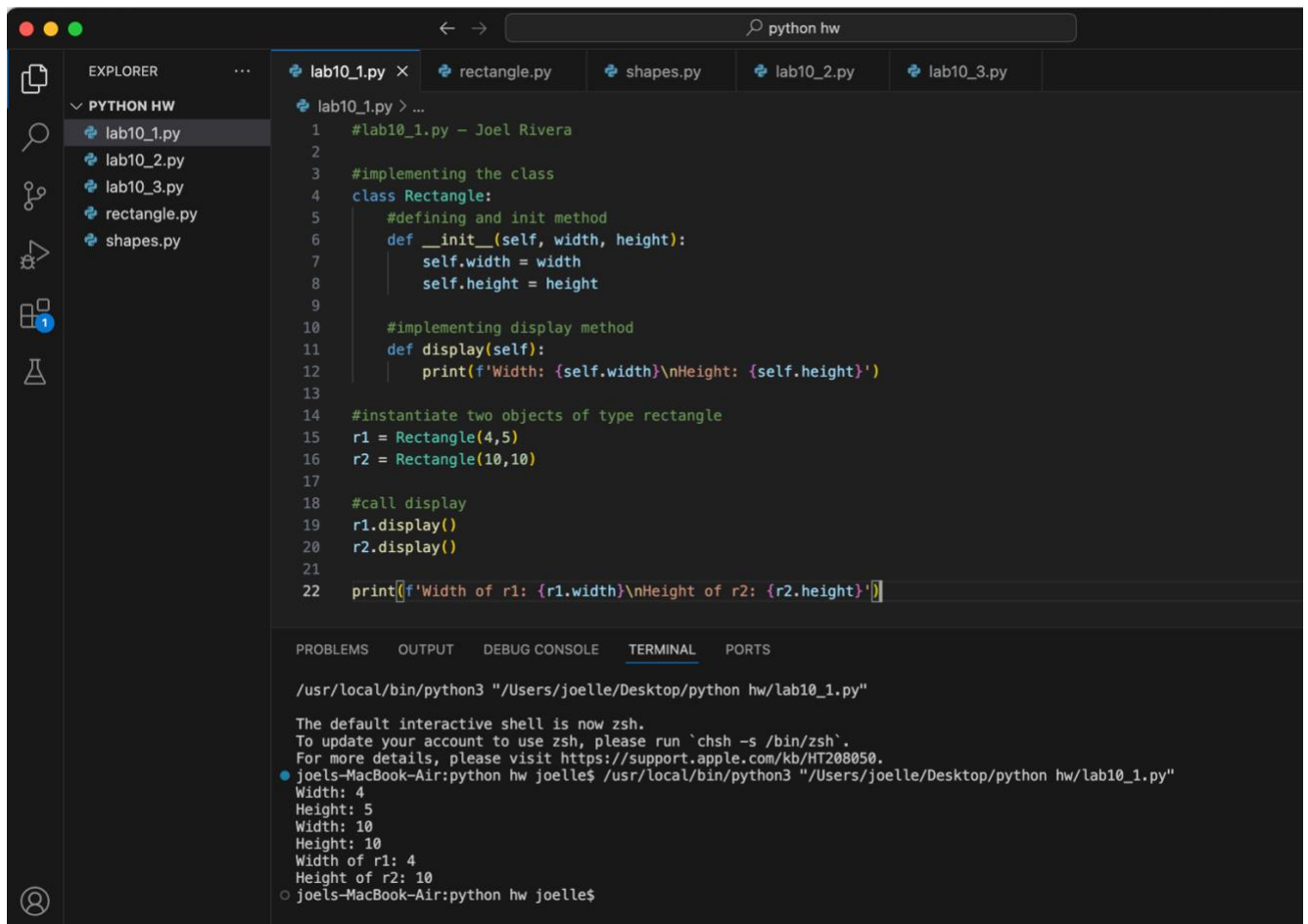


Joel Rivera  
Et-574 F2F4L  
Lab 10  
11/13/2024



```
1  #lab10_1.py - Joel Rivera
2
3  #implementing the class
4  class Rectangle:
5      #defining and init method
6      def __init__(self, width, height):
7          self.width = width
8          self.height = height
9
10     #implementing display method
11     def display(self):
12         print(f'Width: {self.width}\nHeight: {self.height}')
13
14     #instantiate two objects of type rectangle
15     r1 = Rectangle(4,5)
16     r2 = Rectangle(10,10)
17
18     #call display
19     r1.display()
20     r2.display()
21
22     print(f'Width of r1: {r1.width}\nHeight of r2: {r2.height}')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
/usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab10_1.py"

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
● joels-MacBook-Air:python hw joelle$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab10_1.py"
Width: 4
Height: 5
Width: 10
Height: 10
Width of r1: 4
Height of r2: 10
○ joels-MacBook-Air:python hw joelle$
```

The screenshot shows the VS Code editor with the file explorer on the left. The 'PYTHON HW' folder is expanded, showing files: `__pycache__`, `lab10_1.py`, `lab10_2.py` (selected), `lab10_3.py`, `rectangle.py`, and `shapes.py`. The main editor displays the code in `lab10_2.py`:

```
1 #lab10_2.py - Joel Rivera
2 from rectangle import Rectangle
3
4 #instantiate two objects of type rectangle
5 r1= Rectangle(4,5)
6 r2= Rectangle()
7
8 #calling display(), printing area()
9 r1.display()
10 print(f'Area: {r1.area()}')
11 r2.display()
12 print(f'Area: {r2.area()}')
13
14 #calling setWidth(), setHeight() and updating value
15 r2.setWidth(6)
16 r2.setHeight(6)
17
18 #displaying getWidth(), getHeight() with print
19 print(f'Get Width: {r2.getWidth()}')
20 print(f'Get Height: {r2.getHeight()}')
21
22 #printing area() of r2
23 print(f'Area: {r2.area()}')
```

The terminal at the bottom shows the output of the script:

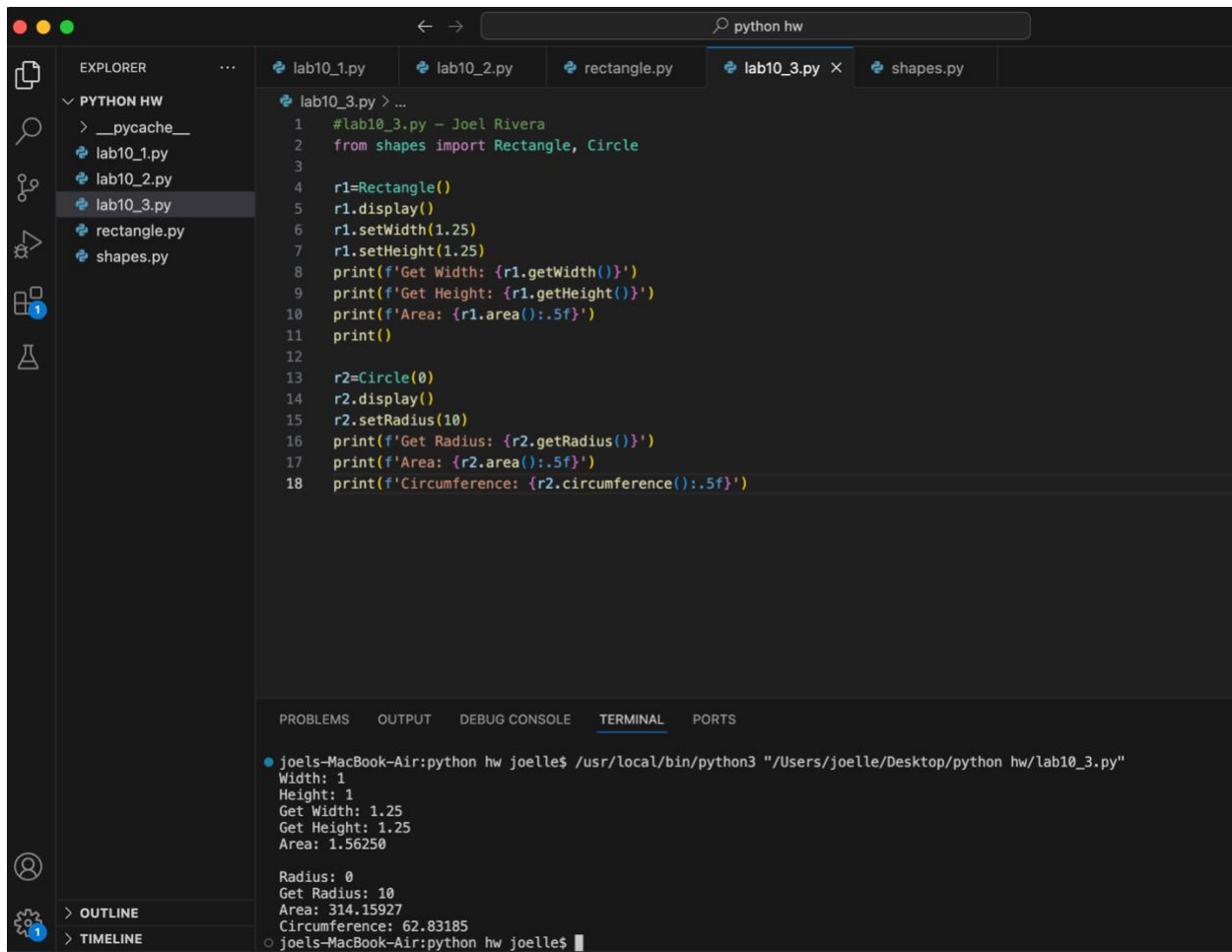
```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
joels-MacBook-Air:python hw joelles$ /usr/local/bin/python3 "/Users/joelle/Desktop/python hw/lab10_1.py"
Width: 4
Height: 5
Area: 20
Width: 1
Height: 1
Area: 1
Get Width: 6
Get Height: 6
Area: 36
joels-MacBook-Air:python hw joelles$
```

The screenshot shows the VS Code editor with the file explorer on the left. The 'PYTHON HW' folder is expanded, showing files: `__pycache__`, `lab10_1.py`, `lab10_2.py`, `rectangle.py` (selected), `lab10_3.py`, and `shapes.py`. The main editor displays the code in `rectangle.py`:

```
1 #lab10_2.py - Joel Rivera
2
3 #implement class rectangle
4 class Rectangle:
5     #implement init method with optional parameter type
6     def __init__(self, width=1, height=1, type=None):
7         self.width= width
8         self.height= height
9         self.type= type
10
11 #implement display()
12 def display(self):
13     print(f'Width: {self.width}\nHeight: {self.height}')
14
15 #implement setwidth() to assign width to the instance variable
16 def setWidth(self, width):
17     self.width= width
18
19 #implement setheight() to assign width to the instance variable
20 def setHeight(self, height):
21     self.height= height
22
23 #implement getWidth() to return value instance variable width
24 def getWidth(self):
25     return self.width
26
27 #implement getHeight() to return value instance variable height
28 def getHeight(self):
29     return self.height
30
31 #implement area() to return value of area of rectangle
32 def area(self):
33     return self.width * self.height
```

The terminal at the bottom shows the output of the script:

```
Area: 1
Get Width: 6
Get Height: 6
Area: 36
joels-MacBook-Air:python hw joelles$
```



The screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'PYTHON HW' containing files: '\_\_pycache\_\_', 'lab10\_1.py', 'lab10\_2.py', 'lab10\_3.py', 'rectangle.py', and 'shapes.py'. The 'shapes.py' file is open in the editor, showing the implementation of a 'Rectangle' class. The code includes an initialization method with optional parameters, display, setWidth, setHeight, getWidth, getHeight, and area methods. A 'Circle' class is also partially visible at the bottom.

```
1 #lab10_3.py - Joel Rivera
2 from math import pi
3
4 #implement class rectangle
5 class Rectangle:
6     #implement init method with optional parameter type
7     def __init__(self, width=1, height=1, type=None):
8         self.width= width
9         self.height= height
10        self.type= type
11
12    #implement display()
13    def display(self):
14        print(f'Width: {self.width}\nHeight: {self.height}')
15
16    #implement setWidth() to assign width to the instance variable
17    def setWidth(self, width):
18        self.width= width
19
20    #implement setHeight() to assign width to the instance variable
21    def setHeight(self, height):
22        self.height= height
23
24    #implement getWidth() to return value instance variable width
25    def getWidth(self):
26        return self.width
27
28    #implement getHeight() to return value instance variable height
29    def getHeight(self):
30        return self.height
31
32    #implement area() to return value of area of rectangle
33    def area(self):
34        return self.width * self.height
35
36
37    #implement circle class
38    class Circle:
39        #implement __init__() with radius and optional parameter type
40        def __init__(self, radius=1, type=None):
```

The screenshot shows the VS Code editor with the file explorer on the left displaying the same project. The 'shapes.py' file is open, showing the implementation of a 'Circle' class. The code includes an initialization method with optional parameters, display, setRadius, getRadius, area, and circumference methods.

```
5 class Rectangle:
6     #implement init method with optional parameter type
7     def __init__(self, width=1, height=1, type=None):
8         self.width= width
9         self.height= height
10        self.type= type
11
12    #implement display()
13    def display(self):
14        print(f'Width: {self.width}\nHeight: {self.height}')
15
16    #implement setWidth() to assign width to the instance variable
17    def setWidth(self, width):
18        self.width= width
19
20    #implement setHeight() to assign width to the instance variable
21    def setHeight(self, height):
22        self.height= height
23
24    #implement getWidth() to return value instance variable width
25    def getWidth(self):
26        return self.width
27
28    #implement getHeight() to return value instance variable height
29    def getHeight(self):
30        return self.height
31
32    #implement area() to return value of area of rectangle
33    def area(self):
34        return self.width * self.height
35
36
37    #implement circle class
38    class Circle:
39        #implement __init__() with radius and optional parameter type
40        def __init__(self, radius=1, type=None):
41            self.radius= radius
42            self.type= type
43
44        #implement display() to print value of radius
45        def display(self):
46            print(f'Radius: {self.radius}')
47
48        #implement setRadius() to assign radius to the instance variable
49        def setRadius(self, radius):
50            self.radius= radius
51
52        #implement getRadius() to return the value of instance variable radius
53        def getRadius(self):
54            return self.radius
55
56        #implement area() to return the value of area of a circle
57        def area(self):
58            return pi * self.radius**2
59
60        #implement circumference(), return circumference of a circle.
61        def circumference(self):
62            return 2 * pi * self.radius
```

4. I faced more than two challenges in this week's lab lol but the first main challenge was in question one; within the display method I was originally passing width and not getting

the desired output but an error. I realized that I needed to use `self` since it refers to the current instance of `self.width`. when I made the change, I got the desired output. The second challenge came from the second question; when I was trying to define the `setWidth()` method I had originally wrote `width= self.width` and the code wasn't working but when I switch it around to then assign the value of `width` to the instance variable `self.width`, I got the desired output.