

Exercise 17.11.2

Exercise_17.11.2

class Point:

def __init__(self, initX, initY):

self.x = initX

self.y = initY

def getX(self):

return self.x

def getY(self):

return self.y

def reflect_x(self):

reflects the point across x-axis

newY = - self.y

return(self.x, newY)

point1 = Point(7, 11)

point2 = Point(-2, 3)

point3 = Point(1.23, -9.876)

print(point1.reflect_x())

print(point2.reflect_x())

print(point3.reflect_x())

Initial input	Result
<pre>point1 = Point(7, 11) point2 = Point(-2, 3) point3 = Point(1.23, -9.876)</pre>	<pre>==== RESTART: /U (7, -11) (-2, -3) (1.23, 9.876)</pre>

Exercise 17.11.4

Exercise_17.11.4

import math

class Point:

def __init__(self, initX, initY):

self.x = initX

self.y = initY

def getX(self):

return self.x

def getY(self):

return self.y

def get_line_to(self, other_point):

m_num = abs(self.y - other_point.y)

m_denom = abs(self.x - other_point.x)

if m_denom == 0:

return("Undefined Line")

```

else:
    m = abs((self.y - other_point.y) / (self.x - other_point.x))
    b = self.y - (m * self.x)
    return(m, b)

```

```
point1 = Point(4, 11)
```

```
point2 = Point(6, 15)
```

```
point3 = Point(3, 1)
```

```
point4 = Point(3, 5)
```

```
point5 = Point(5, 4)
```

```
point6 = Point(55, 11)
```

```
print(point1.get_line_to(point2))
```

```
print(point3.get_line_to(point4))
```

```
print(point5.get_line_to(point6))
```

Initial input	Result
<pre> point1 = Point(4, 11) point2 = Point(6, 15) point3 = Point(3, 1) point4 = Point(3, 5) point5 = Point(5, 4) point6 = Point(55, 11) print(point1.get_line_to(point2)) print(point3.get_line_to(point4)) print(point5.get_line_to(point6)) </pre>	<pre> == RESTART: /Us (2.0, 3.0) Undefined Line (0.14, 3.3) </pre>

Exercise 18.6.8

Exercise_18.6.8

class Point:

def __init__(self, initX, initY):

self.x = initX

self.y = initY

def getX(self):

return self.x

def getY(self):

return self.y

def __str__(self):

return "(" + str(self.x) + ", " + str(self.y) + ")"

class Rectangle:

def __init__(self, initP, initW, initH):

self.point = initP

self.width = initW

self.height = initH

def getWidth(self):

return self.width

```
def getHeight(self):
    return self.height
```

```
def __str__(self):
    return "Point = " + str(self.point) + ", Width = " + str(self.width) + ", Height = " + str(self.height)
```

```
start_point = Point(2, 3)
print(Rectangle(start_point, 5, 7))
```

Initial input	Result
<pre>start_point = Point(2, 3) print(Rectangle(start_point, 5, 7))</pre>	<pre>cnous.py Point = (2, 3), Width = 5, Height = 7</pre>

Exercise 18.6.10

Exercise_18.6.10

```
class Point:
```

```
    def __init__(self, initX, initY):
        self.x = initX
        self.y = initY
```

```
    def getX(self):
        return self.x
```

```
    def getY(self):
        return self.y
```

```
def __str__(self):
    return "(" + str(self.x) + ", " + str(self.y) + ")"
```

```
class Rectangle:
```

```
    def __init__(self, initP, initW, initH):
        self.point = initP
        self.width = initW
        self.height = initH
```

```
    def getWidth(self):
        return self.width
```

```
    def getHeight(self):
        return self.height
```

```
        def __str__(self):
            return "Point = " + str(self.point) + ", Width = " + str(self.width) + ", Height = " + str(self.height)
```

```
    def perimeter(self):
        return (self.width * 2) + (self.height * 2)
```

```
r = Rectangle(Point(0, 0), 10, 5)
print(r.perimeter())
```

Initial input	Result
<pre>r = Rectangle(Point(0, 0), 10, 5) print(r.perimeter())</pre>	<pre>30</pre>

