

# SelfEvaluation12\_5

November 15, 2015

## 1 Exercise 12.5:

Design a simple mapping system class hierarchy in Python. Call the base class **Shape** and the child classes are **Point**, **Polyline** and **Polygon**. The base class has a method **draw** and each child class has its own **draw** method. You don't need to implement the draw method. Just provide the program outline using pass statements as placeholders. Feel free to add class members and even other classes you think are necessary.

*Answer appears after one blank page (so you don't peek).*

Are you sure you're ready to peek?

## 2 Possible Solution

This is the simplest solution (with a bit of error checking):

```
In [38]: class Shape:
    # Shape doesn't really do anything.
    def __init__(self, type = 'Unknown'):
        self.type = type

    def draw(self):
        pass

    def getType(self):
        return self.type

class Point(Shape):
    def __init__(self, x = -999., y = -999.):
        # Each point pair should be a tuple.
        Shape.__init__(self, type = 'Point')
        self.x = x
        self.y = y

    def draw(self):
        pass

class PolyLine(Point):
    def __init__(self, vertices = [], x = [-999, -999], y = [-999,-999]):
        Shape.__init__(self, "PolyLine")
        # Make sure that there's at least two points:
        if len(vertices) == 0:
            if len(x) == len(y) > 1:
                self.vertices = []
                for i in xrange(len(x)):
                    self.vertices.append((x[i], y[i]))
            else:
                raise Exception("PolyLine requires at least two points.")
        else:
            if len(vertices) == 1:
                raise Exception("PolyLine requires at least two points.")
            else:
                self.vertices = vertices

    def draw(self):
        pass

class Polygon(Point):
    def __init__(self, vertices = [], x = [-999, -999,-999], y = [-999,-999,-999]):
        Shape.__init__(self, "Polygon")
        # Make sure that there's at least three points:
        if len(vertices) == 0:
            if len(x) == len(y) > 2:
                self.vertices = []
                for i in xrange(len(x)):
                    self.vertices.append((x[i], y[i]))
```

```
        else:
            raise Exception("Polygon requires at least two points.")
    else:
        if len(vertices) < 3:
            raise Exception("Polygon requires at least two points.")
        else:
            self.vertices = vertices

    def draw(self):
        pass

raw_shape = Shape('YesName')
raw_point = Point()
raw_polyline = PolyLine()
raw_polygon = Polygon()

print raw_shape
print raw_point
print raw_polyline
print raw_polygon
```

```
<__main__.Shape instance at 0x030C8A58>
<__main__.Point instance at 0x030C8EE0>
<__main__.PolyLine instance at 0x030C88F0>
<__main__.Polygon instance at 0x030C8300>
```