

```

1 class Point:
2     """ Point class for representing and manipulating x,y coordinates.
3
4     def __init__(self):
5         """ Create a new point at the origin """
6         self.x = 0
7         self.y = 0
8
9 p = Point()      # Instantiate an object of type Point
10 q = Point()     # and make a second point
11
12 print("Nothing seems to have happened with the points")
13

```

Nothing seems to have happened with the points

```

1 class Point:
2     """ Point class for representing and manipulating x,y coordinates.
3
4     def __init__(self):
5         """ Create a new point at the origin """
6         self.x = 0
7         self.y = 0
8
9 p = Point()      # Instantiate an object of type Point
10 q = Point()     # and make a second point
11
12 print(p)
13 print(q)
14
15 print(p is q)
16

```

```

<__main__.Point object>
<__main__.Point object>
False

```

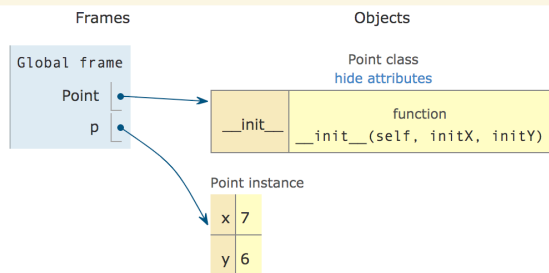
```

1 class Point:
2     """ Point class for representing and manipulating x,y coordinates.
3
4     def __init__(self, initX, initY):
5         """ Create a new point at the given coordinates """
6         self.x = initX
7         self.y = initY
8
9 → p = Point(7, 6)

```

→ line that has just executed
 → next line to execute

Visualized using Online Python Tutor by Philip Guo



```

1 class Point:
2     """ Point class for representing and manipulating x,y coordinates.
3
4     def __init__(self, initX, initY):
5         """ Create a new point at the given coordinates. """
6         self.x = initX
7         self.y = initY
8
9     def getX(self):
10        return self.x
11
12    def getY(self):
13        return self.y
14
15
16 p = Point(7, 6)
17 print(p.getX())
18 print(p.getY())
19

```

7
6

```

1 class Point:
2     """ Point class for representing and manipulating x,y coordinates.
3
4     def __init__(self, initX, initY):
5         """ Create a new point at the given coordinates. """
6         self.x = initX
7         self.y = initY
8
9     def getX(self):
10        return self.x
11
12    def getY(self):
13        return self.y
14
15    def distanceFromOrigin(self):
16        return ((self.x ** 2) + (self.y ** 2)) ** 0.5
17
18
19 p = Point(7, 6)
20 print(p.distanceFromOrigin())
21

```

9.21954445729

Run

Show CodeLens

```
1 import math
2
3 class Point:
4     """ Point class for representing and manipulating x,y coordinates
5
6     def __init__(self, initX, initY):
7         """ Create a new point at the given coordinates. """
8         self.x = initX
9         self.y = initY
10
11     def getX(self):
12         return self.x
13
14     def getY(self):
15         return self.y
16
17     def distanceFromOrigin(self):
18         return ((self.x ** 2) + (self.y ** 2)) ** 0.5
19
20 def distance(point1, point2):
21     xdiff = point2.getX() - point1.getX()
22     ydiff = point2.getY() - point1.getY()
23
24     dist = math.sqrt(xdiff**2 + ydiff**2)
25
```

5.0

Run

Show CodeLens

```
1 class Point:
2     """ Point class for representing and manipulating x,y coordinates.
3
4     def __init__(self, initX, initY):
5         """ Create a new point at the given coordinates. """
6         self.x = initX
7         self.y = initY
8
9     def getX(self):
10        return self.x
11
12    def getY(self):
13        return self.y
14
15    def distanceFromOrigin(self):
16        return ((self.x ** 2) + (self.y ** 2)) ** 0.5
17
18
19 p = Point(7, 6)
20 print(p)
21
```

<__main__.Point object>

Run

Show CodeLens

```
1 class Point:
2     """ Point class for representing and manipulating x,y coordinates.
3
4     def __init__(self, initX, initY):
5         """ Create a new point at the given coordinates. """
6         self.x = initX
7         self.y = initY
8
9     def getX(self):
10        return self.x
11
12    def getY(self):
13        return self.y
14
15    def distanceFromOrigin(self):
16        return ((self.x ** 2) + (self.y ** 2)) ** 0.5
17
18    def __str__(self):
19        return "x=" + str(self.x) + ", y=" + str(self.y)
20
21 p = Point(7, 6)
22 print(p)
23
```

x=7, y=6

Run

Show CodeLens

```
1 class Point:
2
3     def __init__(self, initX, initY):
4         """ Create a new point at the given coordinates. """
5         self.x = initX
6         self.y = initY
7
8     def getX(self):
9         return self.x
10
11    def getY(self):
12        return self.y
13
14    def distanceFromOrigin(self):
15        return ((self.x ** 2) + (self.y ** 2)) ** 0.5
16
17    def __str__(self):
18        return "x=" + str(self.x) + ", y=" + str(self.y)
19
20    def halfway(self, target):
21        mx = (self.x + target.x) / 2
22        my = (self.y + target.y) / 2
23        return Point(mx, my)
24
25 p = Point(2, 4)
```

```
x=4.0, y=8.0
4.0
8.0
```

