

Name: \_\_\_\_\_ CM: \_\_\_\_\_ Section: \_\_\_\_\_ Grade: \_\_\_\_\_ of 10

Here (below) is a partial definition and test code for a simple **Point** class (as you saw/worked in a previous session).

<pre># Tests the Point class p1 = Point(40, 50) p2 = Point(60, 70) print(p1, p2)  p2.move_by(1, 2) print(p1, p2)  a = p1.get_number_of_moves() b = p2.get_number_of_moves() print(a, b)</pre>	<pre>class Point(object):     def __init__(self, x, y):         self.x = x         self.y = y         self.total_moves = _____      def move_by(self, dx, dy):         # Location 1         self.x = self.x + dx         self.y = self.y + dy         self.total_moves = self.total_moves + 1      def __repr__(self):         return "Point({}, {})".format(_____,  _____)      def get_number_of_moves(self):         # Location 2         return total_moves</pre>
---	---

1. We want the `__repr__` method to print the current **x** and **y** coordinates of its **Point**, formatted nicely. Fill in the blanks above in `__repr__` to make it do that.
2. In the space to the right, draw a box-and-point diagram that shows the values of **p1**, **p2**, and **self** when the code gets to **Location 1**.
3. Fill in the blank in `__init__` to set **self.total\_moves** to its correct value.
4. There is a small but important bug inside the **get\_number\_of\_moves** method. What is it?
5. When the test code runs and gets to **Location 2** the FIRST time, what is the value of **self**?  
What is the value of **self** when we get to Location 2 the SECOND time?
6. Assume that all the code works as intended (that is, assume that the bug in **get\_number\_of\_moves** is fixed). In the space to the right, show the output of the test code.
7. Are you very, very confident that you know **what lines of code execute, in what order**, when the test code runs? That you **understand what self is and why its use `** attaches data **` to Point objects?**  
**Yes No** (if *No*, then talk with an assistant or your instructor about this quiz).