

Test 2 – Practice Problems for the Paper-and-Pencil portion

1. In Session 8, you implemented a *Point* class. Recall that a *Point* object has instance variables *x* and *y* for its x and y coordinates.

Consider the code in the box below. On the **next** page, draw the **box-and-pointer diagram** for what happens when *main* runs. Also on the next page, show what the code would **print** when *main* runs.

```
def main():
    point1 = Point(8, 10)
    point2 = Point(20, 30)
    x = 405
    y = 33

    print('Before:', point1, point2, x, y)

    z = change(point1, point2, x, y)

    print('After:', point1, point2, x, y, z)

def change(point1, point2, x, a):
    point2.x = point1.x
    point2 = Point(5, 6)
    point1.y = point2.y
    x = 99
    point1.x = x
    a = 77

    print('Within:', point1, point2, x, a)

    return a
```

Draw your box-and-pointer diagram here:

What prints when *main* runs?

Assume that *Point* objects get printed as per this example: **Point(8, 10)**.

Before: _____

Within: _____

After: _____

2. In Session 8, you implemented a ***Point*** class. Recall that a ***Point*** object has instance variables ***x*** and ***y*** for its x and y coordinates.

Here, you will implement a portion of a class called ***Triangle***, described as follows:

- The ***Triangle*** constructor takes 3 arguments, each a ***Point*** object: one for each corner of the ***Triangle***.
- The ***Triangle*** class has a method called ***spin(x, y)***. It changes the first of the Triangle's corners to ***(x, y)***, where ***x*** and ***y*** are arguments of the ***spin*** method. It also swaps the other two of the Triangle's corners.

For purposes of this problem, you can treat whichever corner you wish as the “first” of the Triangle's corners. Also, we don't care in this problem whether you construct new Points or make variables refer to existing ones.

Write the implementation of the class here:

For each of the following, write one TEST of:

- The ***construction*** of Triangle objects.

- The execution of the ***spin*** method.