

Name: _____ CM: _____ Section: _____ Grade: _____ of 10

```
class SecretAgent(object):
    """ A secret agent w/ multiple identities. """
    def __init__(self, names):
        self.names = names
        self.current_name_index = 0

    def show_all_names(self):
        """ Prints all of this SecretAgent's names,
            each on their own line. """

    def change_to_next_name(self):
        # --- Location 1 ---
        self.current_name_index = self.current_name_index + 1

    def give_current_name(self):
        """ Returns the name that this Secret Agent is currently using. """

agent1 = SecretAgent(["Mary", "Jane", "Lola", "Terri"])
agent2 = SecretAgent(["Bob", "Niko", "Rick", "Zane", "Alec"])
agent1.change_to_next_name()
print(agent1.give_current_name())
print(agent2.give_current_name())
```

1. Fill in the code for `show_all_names`.
2. Fill in the code for `give_current_name`. It must use the `current_name_index`.
3. Which object is *self* when Location 1 is executed in the above code? _____
4. (a) If you call `change_to_next_name()` repeatedly, it works for a while, but what error will you eventually get and why?

(b) Instead, once you get to the last name, it should cycle back to the first name. Add code to that method above to make that happen and so fix the bug.
5. What is the output of running the completed code above?

6. Consider the code snippet below. It is a contrived example with poor style, but it will run without errors. What does it print when it runs?

Write your answer in the box to the right of the code.

Showing your work by marking up the code to show its execution is the best way to allow for partial credit.

```
def main():
    z = one(3)
    print('Main:', one(two(2)), z)

def one(x):
    print('One:', x)
    y = two(10 * x)
    x = x + 7
    y = y + two(x)
    return y
    print('Here:', x)
    return y

def two(y):
    print('Two:', y)
    return y + 5

main()
```

Output:

7. Consider the code snippet to the right. It is a contrived example with poor style, but it will run without errors.

What does it print when it runs? Write your answer in the box below. We suggest that you use the space on a separate page to keep track of the values of **k**, **a**, **b** and **s** as you work.

```
a = 1
b = 4
s = [8, 6, 20, 10, 30, 40, 50]
for k in range(b):
    a = a + s[k + 1]
    s[len(s) - 1 - k] = k
    print(k, a, s)
```

Output:

8. Consider the code snippet to the right. It is a contrived example with poor style, but it will run without errors.

Draw a *box-and-pointer diagram* for the execution of the code. Then show, in the box below, what the code *prints* when it runs.

Output:

```
p1 = rg.Point(6, 0)
p2 = rg.Point(4, 5)
p3 = rg.Point(10, 8)
p4 = p1
p1 = p3
p1.x = 99
p2.x = 707
p3.y = 33
p4.y = 200
p3 = rg.Point(12, p3.x)
p3.x = 55

print(p1.x, p1.y)
print(p2.x, p2.y)
print(p3.x, p3.y)
print(p4.x, p4.y)
```

Draw your *box-and-pointer diagram* below here, or use a separate sheet (your choice):

9. Consider the code on the next page. It is a contrived example with poor style but will run without errors. In this problem, you will trace the execution of the code. As each location is encountered during the run: **CIRCLE** each variable that is *defined* at that location; **WRITE** the **VALUE** of each variable that you *circled* directly **BELOW** the circle.

Location 1 (1st time)	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 2 (1st time)	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 1 (2nd time)	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 2 (2nd time)	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 3	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 4	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 5	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 6	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 7	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b
Location 8	a	z	self.a	self.b	t1.a	t1.b	t2.a	t2.b	r.a	r.b	s.a	s.b

ASK FOR HELP IF YOU DO NOT UNDERSTAND WHAT THIS PROBLEM ASKS YOU TO DO.

Use the space to the right or a separate sheet to show your work in any way that you find helpful. We suggest a Box and Pointer diagram.

```
class Thing(object):
    def __init__(self, a, z):
        ##### --- Location 1 ---
        self.a = a * 10
        self.b = z
        a = 99
        ##### --- Location 2 ---

    def go(self, r, s):
        ##### --- Location 3 ---
        self.a = 13
        self.b = 70
        ##### --- Location 4 ---
        r.a = 91
        s.b = 11
        z = 44
        ##### --- Location 5 ---
        return 88

def main():
    a = 3
    z = 100
    t1 = Thing(4, z)
    ##### --- Location 6 ---
    t2 = Thing(22, 505)
    ##### --- Location 7 ---
    z = t1.go(t1, t2)
    ##### --- Location 8 ---
```

main()