Exam 1 – Paper and Pencil – In-class Practice

Name: ______________________________ **Section:** ________

1. 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. Show your work by making notations in the code or by using the empty space below or on another sheet of paper, as desired.

```
def main():
    a = 3
    b = 1
               3  1
    a = two(a, b)   460
    print("Main:", a, b)

        3  1
def two(x, a):
    print("Two 1:", x, a)
                  4
    z = one(one(x + 1, 10), 2)  455
                    45
    print("Two 2:", z)
    return z + 5   460
    z = z + 20
    return z

        ~~1 2~~ ~~4 10~~ 45 2
def one(b, a):
    print("One 1:", a, b)
    b = b * 10    ~~10~~ ~~40~~ 450
    print("One 2:", a, b)
    return b + 5   ~~15~~ ~~45~~ 455


                         15
print("Bottom:", one(1, 2))
main()
```

Output:

```
One 1: 2 1
One 2: 2 10
Bottom: 15
Two 1: 3 1
One 1: 10 4
One 2: 10 40
One 1: 2 45
One 2: 2 450
Two 2: 455
Main: 460 1
```

Your name: SOLUTION

2. Consider the code below. 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:

    1. ***CIRCLE*** **each name (i.e., variable)** that is ***defined*** at that Location.
    2. ***WRITE*** the ***VALUE*** of each name (i.e., variable) that you ***circled*** directly ***BELOW*** the circle.

Note that you fill in the table in the order that Locations are encountered, NOT from top to bottom. Ask for help if you do not understand these instructions.

```
def main():
    #### Location 1
    a = 1
    b = 2
    c = 3
    d = 4
    e = 5
    #### Location 2

                4  2  3  1
    a = boo(d, b, c, a) 4
    #### Location 3


         4  2  3  1
def boo(a, b, c, d):
    #### Location 4
    b = 10
    r = a + 10  14
    #### Location 5
    return a  4


#### Location 6
main()
#### Location 7
```

Location 1	a	b	c	d	e	r
Location 2	[✓ a] 1	[✓ b] 2	[✓ c] 3	[✓ d] 4	[✓ e] 5	r
Location 3	[✓ a] 4	[✓ b] 2	[✓ c] 3	[✓ d] 4	[✓ e] 5	r
Location 4	[✓ a] 4	[✓ b] 2	[✓ c] 3	[✓ d] 1	e	r
Location 5	[✓ a] 4	[✓ b] 10	[✓ c] 3	[✓ d] 1	e	[✓ r] 14
Location 6	a	b	c	d	e	r
Location 7	a	b	c	d	e	r

~~4~~3. Assume that you have names (i.e, variables) **a** and **b** that are positive integers with **b > 2**. In the box below, ***fill in the blanks*** with code that would print every integer from **a + 2** to **a + b** including the **a + b** value. For example, if **a = 7** and **b = 5**, your code would print every integer from **9** to **12**, thus producing the output shown to the right.

```
9
10
11
12
```

Your code must be for the ***generic case*** for **a** and **b**. That is, use the names (i.e., variables) **a** and **b** in filling in the blanks, as appropriate. As in all problems throughout Exam 1, you must use the ***single-argument*** form of **range**, as in **range(blah)**. You may NOT use the multiple-argument form of **range**, as in **range(r, s)**.

```
for k in range(____(a+b) - (a+2) + 1____):

    print(____K + (a+2)____)
```

4. 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 below and to the right.

Suggestion: Use a table on scratch paper to keep track of the values of **k, a** and **b** as you trace through the code.

```
a = 2
b = 1
for k in range(4):
    a = a + 2
    b = a + b + k
    print(k, a, b)

print("ok", a, b)
```

K	a	b
	2	1
0		
	4	5
1	6	12
2	8	22
3	10	35

Output:

```
0  4  5
1  6  12
2  8  22
3  10 35
```