Exam 3 – Paper and Pencil part - Fall, 2020-21

Name: _____ ***SOLUTION and Rubric*** **_____ Section: _____**

Rules and Expectations

At the beginning of this exam, you will receive the **Expectations about Academic Integrity** for this exam -- it is the same as what you were given to read previously. Re-read that document as needed. ***Sign it and turn it in when you <u>finish</u> this exam (both parts).***

Two parts (this is Part 1, Paper-and-Pencil)

For this part, the ONLY external resource you may use is a single 8½ by 11-inch sheet of paper, with whatever you want on it, typed or handwritten or a combination of the two. **You may use BOTH sides of the sheet** (or you may use TWO sheets of paper but using only ONE side of each). You must have prepared the sheet *before* beginning this exam. You may also use a calculator if you like (but only for calculating), along with blank paper.

Problem	Points Possible	Points Earned	Comments
1	**7**		
2	**7**		
3	**16**		
Total (of 100 on the exam)	**30**		

Communication

For both parts of the exam, ***you must not communicate with anyone*** except your instructors and their assistants, if any. In particular:

- You must not talk with anyone else or exchange information with them during this exam.

- After this exam, you must not talk about the exam with anyone who has not yet taken it.

Do NOT use email, *chat* or the like during this exam. **Close any such applications now.**

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. **Showing your work (in any way you wish) is the best way to allow for partial credit.**

```python
r = 10
while True:
    r = r + 3
    if r > 20:
        break
    print(r)
print("A:", r)



total = 0
while total < 8:
    print(total)
    total = total + 2
print("B:", total)
```

Output:

13

16

19

A: 22

not really
any space
here

0

2

4

6

B: 8

Rubric: This problem is worth 7 points. Subtract points as follows:

First part: 4 points

- 4 points for the 4 lines (one point per line of output, but if one line is wrong and subsequent lines are consistent with the error, subtract only the one point).

- Subtract 1 point if there is an extra line(s) of output.

- Don't subtract for obvious arithmetic errors.

- Maximum of -4 (no points) for this part.

Second part: 3 points

- Same rubric as for the first part, but don't subtract more than 3 points for this part.

2. 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 (in any way you wish) is the best way to allow for partial credit.**

```
for j in range(4):
    b = 100
    print(j, b)
    for k in range(j, 3):
        print(j, k, b)
        b = b + j + k
print(b)
```

Output:

I have put extra spaces to make the answer more easily read.

0	100	
0	0	100
0	1	100
0	2	101
1	100	
1	1	100
1	2	102
2	100	
2	2	100
3	100	
100		

Rubric: 7 points, as follows:

- 1 point: The FIRST COLUMN of the **BLUE** lines is correct.

- 1 point: The FIRST COLUMN of the **RED** lines has the same numbers as the blue lines (regardless of how many lines there are).

- 3 points: The SECOND COLUMN of the **RED** lines is correct.

- 2 points: The THIRD COLUMN of the **RED** lines is correct.

Ignore the 100 that should be printed at the end. (Full credit even if missing or wrong.)

Subtract 2 points (once) if there are extra rows.

Throughout, try to subtract only ½ or 1 point for off-by-one errors.

Throughout, try not to "double-count" errors. If they make an error but the subsequent lines are consistent with the error, subtract only for the first error.

If this rubric is unduly harsh, give a "holistic" score. For example, if they seem to understand the basic idea of loops within loops, they should get at least 4 of the 7 points, even if the rubric calls for fewer points.

3. Consider the code snippet below It is a contrived example with poor style but will run without errors.

    On the opposite page, draw a **box-and-pointer diagram** that shows the execution of the code. Then, show **what gets printed** in the indicated box below

Assume that Point and Circle classes are defined as usual for paper-and-pencil problems, where Point objects have instance variables **x** and **y**, and Circle does NOT clone its Point argument, as shown below.

```
class Circle(object):
    def __init__(self, center, r):
        self.center = center
        self.radius = r
```

Assume Point and Circle objects print as in these examples:

   Point(70, 25)   Circle(Point(70, 25), 50)

In the space below, show what gets printed:

580

25

Point(47, 6)

Circle(Point(47, 6), 33)

0   444

1   Point(47, 6)

Rubric: 14 points, as follows:

- Subtract 1 point for each of the 12 numbers that is wrong. But for the second and third times that the Point appears, full credit if they are the same as the first time.

- Subtract up to 3 points for the box-and-pointer diagram, as follows:

    - Full credit if it looks REASONABLE (details of the diagram are UNIMPORTANT for this assessment)

    - 1.5 points if NOT REASONABLE (i.e., there are obvious significant errors) but NOT CRAZY

    - 0 points if CRAZY or MISSING.

```
def main():
    a = 580
    r = 25
    p = Point(a, 88)
    c = Circle(p, r)
    s = [30, p]
    blah(a, p, c, s)

    print(a)
    print(r)
    print(p)
    print(c)
    for k in range(len(s)):
        print(k, s[k])


def blah(a, p, c, s):
    a = 92
    p.x = 47
    c.radius = 33
    c.center.y = 101
    s[0] = 444
    s[1].y = 6
    p = Point(111, 222)


main()
```

Draw your box-and-pointer diagram here: