Test 1 – Paper and Pencil part

Name: ______________________________

Honesty Pledge:

Included in this test is an ***Honesty Pledge*** that is exactly the same as the one which you should have read before the exam. Re-read the Honesty Pledge at the beginning of the exam.

When you are finished with this test, email your instructor saying either:

- I agree with what the Honesty Pledge says, OR
- I ***do NOT agree*** with what the Honesty Pledge says ***and will talk with you privately soon after the test.***

Have you:

- **Successfully completed** and committed ***all the programming exercises from Session 7*****?**
- Checked your ***paper-and-pencil*** **exercises from Session 7** against the answers online?

If not, DO NOT BEGIN THIS EXAM!
Instead, see your instructor to find out what to do.

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. You must have prepared the sheet *before* beginning the exam.

Communication:

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

- You must not talk with anyone else or exchange information with them during the test.
- ***You must NOT use email****, chat* or the like during the test.

Time limit:

You have ***3 hours*** to complete the entire exam – its *paper part* and its *computer part*. Do the paper part first (using only your prepared 1-page-front-and-back sheet). Do not return to the paper part after you begin work on the computer part.

Problem	Points Possible	Points Earned	Comments
1	**2**		
2	**2**		
3	**1**		
4	**2**		
5	**4**		
6	**5**		
7	**3**		
8	**2**		
9	**4**		
Total (of 100 on the test)	**25**		

1. (2 points) True or False:

   As a ***user*** of a function (that is, as someone who will ***call*** the function), you *don't need to know how the function is* ***implemented***; you just need to know the ***specification*** of the function. ***True*** ***False*** (circle your choice)

2. (2 points) What is the value of each of the following expressions?

`21 // 4`	______________________
`21 / 4`	______________________
`21 % 4`	______________________
`4 != 8`	______________________

3. (1 point) What is the value of the following expression?

   `str(3 * 6) + (3 * str(6))` ______________________

   Hint: the *str* function returns a string version of its argument.

4. (2 points) For each of the following Boolean expressions, indicate whether it evaluates to ***True*** or ***False*** (circle your choice):

True	***False***	`(2 > 10)   and   (3 == 3)`
True	***False***	`(2 > 10)   or    (3 == 3)`
True	***False***	`(x > 10)   and   (x < 5)`
True	***False***	`not not True`

5. (4 points) 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.

```
x = 1
for k in range(1, 7, 2):
    x = (2 * x) + (10 * k)
    print(k, x)

print(x)
```

Output:

6. (5 points) Consider the code snippet below. It is a contrived example with poor style, but it will run without errors. What does it print when *main* runs?

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

```
def main():
    first(10)
    second(6)
    third(4)

def first(x):
    print('first')
    print(3 * x)

def second(x):
    print('second')
    if x < 50:
        print(x * x)
    else:
        print(x)

def third(x):
    print('third')
    first(10 * x)
    second(100 * x)
```

Output:

7. (3 points) Consider the code snippet below. It is a contrived example with poor style, but it will run without errors. What does it print when *main* runs?

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

```
def main():
    x = 5
    y = 7
    z = foo(x, y)
    print('main 1:', x, y, z)

    x = 5
    y = 7
    z = foo(y, x)
    print('main 2:', x, y, z)

    z = 1
    z = foo(z, z)
    print('main 3:', x, y, z)

def foo(x, y):
    x = 10
    print('foo:', x, y)
    return (3 * x) + y
```

Output:

8. (2 points)

   a. Write two Python ***constants*** – one an integer (**`int`**) and one a floating point number (**`float`**) – that clearly shows the difference between the **`int`** and **`float`** types.

   b. A Python **`int`** can represent an arbitrarily large number. (circle your choice) ***True*** ***False***

   c. A Python **`float`** can represent an arbitrarily large number. (circle your choice) ***True*** ***False***

   d. There is a limit to the number of significant digits a Python **`float`** can have. (circle your choice) ***True*** ***False***

9. (4 points) Consider a function whose name is ***crazy*** that takes three arguments and prints things per the following example.

```
crazy('OK!', 'Stop', 3)
```

would produce the following output:

```
OK! Stop
Stop OK!
Stop
Stop
Stop
```

That is, the function prints the first argument followed by the second argument, then prints the second argument followed by the first argument. (In each case there is a single space between the two items printed.) Then it prints the second argument the number of times indicated by the third argument.

Write (in the space below) a complete implementation, ***including the header (def) line***, of the ***crazy*** function described above.