Name: _____

Use this quiz to help make sure you understand the videos/reading. ***Answer all questions.*** Make additional notes as desired. ***Not sure of an answer?*** Ask your instructor to explain in class and revise as needed then.

Throughout, where you are asked to "circle your choice", you can circle or underline it (whichever you prefer).

Handout: ***Counted Loops* and *Range Expressions***

1. Write the sequence of numbers that each of the following range expressions generates:

   a. ***range(5)*** – generates the sequence:

   b. **`range(2)`** – generates the sequence:

2. When the code snippet below runs, what gets printed?
   (Show your answer to the right of the box.)

```
for k in range(3):
    print(k, k + 20)
    print((k + 1) * 'hello')
```

3. Write the sequence of numbers that each of the following range expressions generates:

   a. **`range(1, 5)`** – generates the sequence:

   b. **`range(2, 12, 2)`** – generates the sequence:

   c. **`range(12, 2, -2)`** – generates the sequence:

   d. **`range(2, 5)`** – generates the sequence:

   e. **`range(5, 2)`** – generates the sequence:

   f. **`range(5, 2, -1)`** – generates the sequence:

   Hint: The above yield six *different* sequences – none of the answers is the same as any of the other answers.

4. Write a loop that prints `'funny'` 40,000 times.

5. Write a loop that prints the cubes of the numbers from 35 to ***m***, inclusive (where ***m*** is some integer bigger than 35). For example, if ***m*** were 37, then this loop should print:

```
42875      (which is 35 cubed)
46656
50653
```

Textbook Reading: ***Section 1.7 Problem Solving – Algorithm Design*** (pages 16 – 22) and ***Section 2.3 Problem Solving – First Do It By Hand*** (pages 45 – 46).

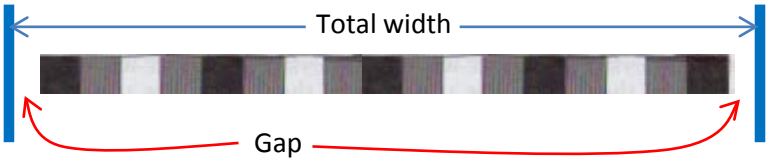
6. Consider the following problem (it is an extension of the problem in Section 2.3):

   A row of black, grey and white tiles needs to be placed along a wall, in a pattern like this:

   For aesthetic reasons, the architect has specified that the first and last tile shall be black.

   Your task is to computer the number of tiles needed, and the gap at each end, given the space available and the width of each tile:

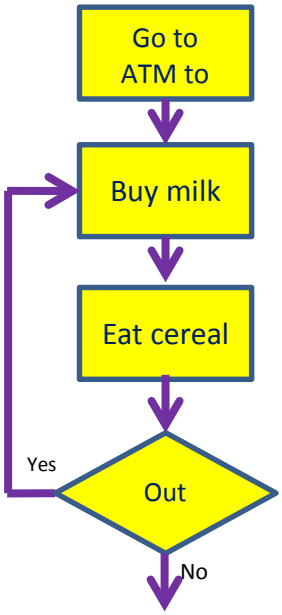
   

   Do a by-hand calculation (using reasonable numbers that you choose) that solves this problem (for the particular numbers you choose). Then, write a formula for the general solution based on your by-hand calculation.

Video: Object Oriented Programming (OOP)

7. Four computer languages developed in the 1950s dominated early computing. One of them is ***Lisp***. What is the name of one of the other 3 languages then?

8. The diagram shown to the right is part of a: (circle your choice)



   a. Poem called *Beowulf*
   b. Rocket ship
   c. Garbage can
   d. Flowchart

9. ***Procedural decomposition*** is: (circle your choice)
   a. The process by which leftovers become fertilizer for gardens
   b. What happens to mummies when they are left out in the open air
   c. How great ideas become hollow ones
   d. The process of breaking a problem into a sequence of subproblems, with each subproblem given a procedure to solve it, and then breaking those subproblems into sub-subproblems, with each sub-subproblem given a procedure to solve it, and so forth until the problem is reduced to procedures of manageable size.

10. The first object-oriented (OO) language was (in some historians' view) SIMULA, but the most influential of the early OO languages was: (circle your choice)
   a. Smalltalk
   b. Bigmouth
   c. Farsight
   d. Giggles

11. *True* or *False*: Most of the widely-used languages developed since 1990 have been ***object oriented*** languages. **True** **False** (circle your choice)

12. The ***procedural*** programming paradigm focuses on: (circle your choice)
    a. Nouns
    b. Verbs
    c. Adjectives

13. The ***object-oriented*** programming paradigm focuses on: (circle your choice)
    a. Nouns
    b. Verbs
    c. Adjectives

14. *True* or *False*: Today's software engineer typically uses *both* the procedural *and* object-oriented paradigms – there is a place for each. **True** **False** (circle your choice)

15. *True* or *False*: Python supports both a procedural and an object-oriented notation. **True** **False** (circle your choice)

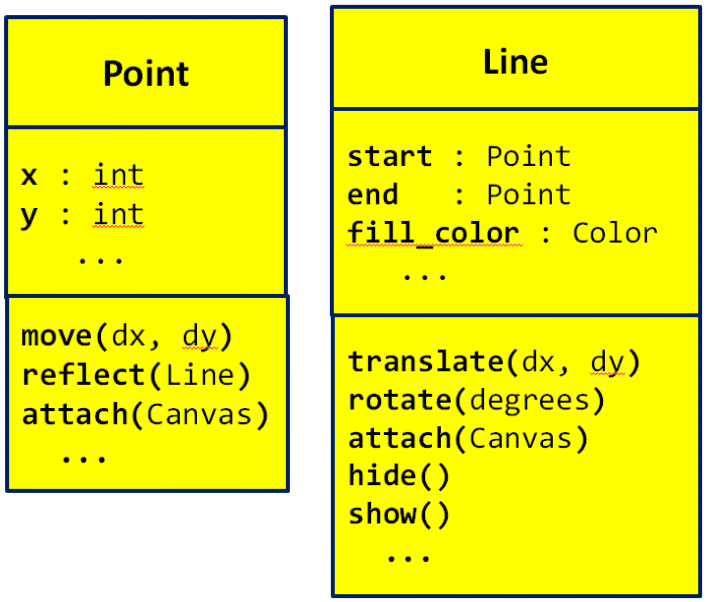
Video: UML Class Diagrams

16. The diagrams to the right are called

    _____ Class Diagrams, where

    _____ stands for Unified Modeling Language. (Fill in both blanks with the (same) 3-letter acronym for ***U***nified ***M***odeling ***L***anguage.)

17. Consider the 2 UML class diagrams shown above and to the right. What are the names of the two ***classes*** shown?



18. Consider the UML class diagram for the ***Point*** class shown above. For that class:
    a. What are the names of the two ***instance variables*** (***fields***) that are shown?

    b. What do you think that those fields represent? (You can't tell this authoritatively from the UML class diagram; just make your best guess based on the names of the fields.)

    c. How many ***methods*** are shown?
    d. How many parameters does the ***move*** method require?
    e. How many parameters does the ***reflect*** method require?
    f. How many parameters does the ***attach*** method require?
    g. What kind of thing is the ***reflect*** method's parameter?
    h. What kind of thing is the ***attach*** method's parameter?

19. Consider the UML class diagram for the ***rg.Line*** class shown above. For that class:
    a. What are the three ***instance variables*** (***fields***) that are shown?

    b. How many ***methods*** are shown?
    c. How many parameters does the ***translate*** method require?
    d. How many parameters does the ***hide*** method require?
    e. What do you think that the ***rotate*** method does? What is its parameter for? (You can't tell this authoritatively from the UML class diagram; just make your best guess based on the names of the method and its parameter.)

Video: Objects and Classes – Using Objects

20. What trick do we use in Eclipse to figure out what **instance variables** and **methods** an object has?

21. What trick do we use in Eclipse to figure out what **arguments** a constructor or method takes?

22. The following examples show two different attempts by two students to construct a new rg.Point at (4, 3). Based on what you learned about constructing objects in Python, which student is correct and why?

```
# Anna
point_anna=rg.Point
point_anna.x=4
point_anna.y=3

# Dalton
point_dalton=rg.Point(0,0)
point_dalton.x=4
point_dalton.y=3
```

23. rg.Point's constructor takes an initial x and a y. Thus, both of the above examples are needlessly long. How would you instead construct the point using just **one** line of code?

24. Based on the video, how would you construct an rg.Circle at the point (100,200) with radius 35 and a blue fill color?

25. Suppose you are told that an rg.Line's constructor takes a start and end point as its constructor arguments. What line of code would you write to create a Line between a point called **point_mine** and a **new Point** at (0, 5)?

Textbook Reading: ***Section 2.4 Strings*** (pages 48 – 51) and ***Sections 2.5.1 – User Input*** and ***2.5.2 – Numerical input*** (pages 55-56).

26. *True* or *False*: One can write string literals like `'Hello'` or like `"Hello"`. That is, one can use single or double quotes to delimit string literals. **True** **False** (circle your choice)

27. Suppose that `s` and `t` are variables whose values are strings. Write a statement that prints the length of the string `s + t + s`

28. Show the output that appears when the following code snippet runs.

```
s = 'Bilbo'
n = len(s)
mystery = s[0] + s[n-1]
print(mystery * 4)
```

29. Consider the following statement:

```
s = input('Enter your name: ')
```

Write a statement that sets the variable `t` to the string that is the same as `s`, except all upper-case. For example, if the human enters *Rhonda*, so that the value of `s` is `'Rhonda'`, then your statement should make `t` have the value `'RHONDA'`. Likewise, if the human enters *gaia*, so that the value of `s` is `'gaia'`, then your statement should make `t` have the value `'GAIA'`.

30. Consider the following statement:

```
s = input('Enter your name: ')
```

Write a statement that sets the variable `t` to the string that is the same as `s`, followed by an upper-case version of `s`, followed by a lower-case version of `s`. For example, if the human enters *Rhonda*, so that the value of `s` is `'Rhonda'`, then your statement should make `t` have the value `'RhondaRHONDArhonda'`.

31. Consider the following statement:

```
s = input('Enter your name: ')
```

Suppose that the human user enters **43** in response to the prompt for input. What will the value of **s + s** be in that case? (Hint: the answer is NOT 86.)

32. Continuing the previous problem, what if the software developer intended that the value of **s + s** be **86** if the user enters **43** as the input (and similarly for other inputs). What should the programmer change or add to the statement above?