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).

***** - **challenge questions.** Think carefully on these and ask about them in class.

Video: Refactoring and Inheritance

1. The following code is an excerpt from a large program of poor quality that prints all prime numbers from 2 to 1000:

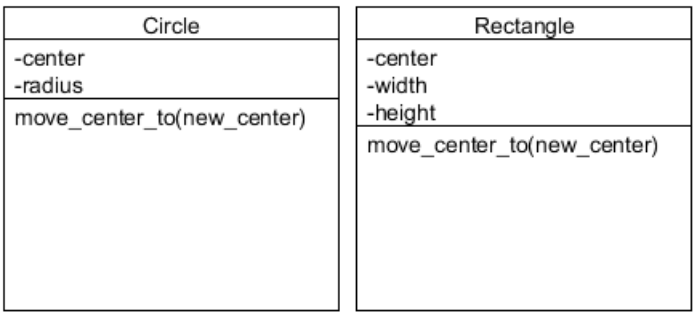
```
def main():
    # find first prime number
    print("prime number 1 is 2")
    # if three is prime, print it.
    nPrimes = 1
    prime = True
    for j in range(2, 3):
        if 3 % j == 0:
            prime = False
            break
    if prime:
        print("prime number {} is {}".format(nPrimes, 3))
        nPrimes = nPrimes + 1

    # if four is prime, print it.
    prime = True
    for j in range(2, 4):
        if 4 % j == 0:
            prime = False
            break
    if prime:
        print("prime number {} is {}".format(nPrimes, 4))
        nPrimes = nPrimes + 1
    # ... etc to 1000
```

   Describe **two** refactoring strategies to reduce the length of this code:

*2. In the space below, apply **both** strategies to refactor the above code.

3. Consider the following UML class diagram describing code of poor quality:



Explain **why** the code implementing this UML diagram would be of poor quality.

*4. Draw an **improved** UML diagram that addresses this issue.

Video: Inheritance – how to use it?

5. Suppose you intend to implement two classes named Vehicle and Car. Write the first line of the Python class declaration that states that a Car is a type of Vehicle.

*6. Suppose a Vehicle takes a VIN in its constructor. Suppose a Car takes a VIN, a make, and a model in its constructor. For instance:

```
vehicle_1=Vehicle(vin_vehicle_1)
car_1=Car(vin_car_1,"Toyota","08-Camry")
```

Write the Car's constructor below, using inheritance to call the superclass constructor: