

Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

GRADE
100%

Practice Quiz: Object-oriented Programming (Optional)

TOTAL POINTS 2

1. Let's test your knowledge of using dot notation to access methods and attributes in an object. Let's say we have a class called Birds. Birds has two attributes: color and number. Birds also has a method called count() that counts the number of birds (adds a value to number). Which of the following lines of code will correctly print the number of birds? Keep in mind, the number of birds is 0 until they are counted! 1 / 1 point

☐ bluejay.number = 0

print(bluejay.number)

☐ print(bluejay.number.count())

☒ bluejay.count()

print(bluejay.number)

☐ print(bluejay.number)

✓ **Correct**

Nice job! We must first call the count() method, which will populate the number attribute, allowing us to print number and receive a correct response.

2. Creating new instances of class objects can be a great way to keep track of values (called attributes, remember? Though attributes do not have to be a value, they can be strings or anything else) associated with the object. This makes it easier to add and subtract from values associated with the objects in a class. The following code illustrates a famous quote by George Bernard Shaw, using objects to represent people. Fill in the blanks to make the code fulfill the behavior described in the quote.

```
1 # "If you have an apple and I have an apple and we exchange these apples, then
```

```

2 # you and I will still each have one apple. But if you have an idea and I have
3 # an idea and we exchange these ideas, then each of us will have two ideas."
4 # George Bernard Shaw
5
6 class Person:
7     apples = 0
8     ideas = 0
9
10 johanna = Person()
11 johanna.apples = 1
12 johanna.ideas = 1
13
14 martin = Person()
15 martin.apples = 2
16 martin.ideas = 1
17
18 def exchange_apples(you, me):
19     # "you" and "me" will exchange ALL our apples with one another
20     x = me.apples
21     me.apples = you.apples
22     you.apples = x
23     return you.apples, me.apples
24
25 def exchange_ideas(you, me):
26     # "you" and "me" will share our ideas with one another
27     you.ideas += me.ideas
28     me.ideas = you.ideas
29     return you.ideas, me.ideas
30
31 exchange_apples(johanna, martin)
32 print("Johanna has {} apples and Martin has {} apples".format(johanna.apples,
33     martin.apples))
34 print("Johanna has {} ideas and Martin has {} ideas".format(johanna.ideas,
35     martin.ideas))

```

Run

Reset

Johanna has 2 apples and Martin has 1 apples
 Johanna has 2 ideas and Martin has 2 ideas



Correct

Awesome! You're getting used to using instances of class objects and assigning them attributes!