Back to fopp

course_4_assessment_1

Due: 2019-02-04 15:12:00

Description: Assessment for the Classes lesson

Score: 0 of 3 = 0.0%

Questions

Not yet graded

Define a class called `Bike` that accepts a string and a float as input, and assigns those inputs respectively to two instance variables, `color` and `price`. Assign to the variable `testOne` an instance of `Bike` whose color is **blue** and whose price is **89.99**. Assign to the variable `testTwo` an instance of Bike whose color is **purple** and whose price is **25.0**.

Save & Run

5/15/2021, 1:08:02 PM - 2 of 2

Show in CodeLens

```
class Bike:
    """ Point class for representing and manipulating x,y coordinates. """

    def __init__(self, initX, initY):

        self.color = initX
        self.price = initY

testOne = Bike('blue',89.99)
testTwo = Bike('purple',25.0)


```

ActiveCode (ac_ch13_01)

Result	Actual Value	Expected Value	Notes
Pass	'blue'	'blue'	Testing that testOne has the correct color assigned.
Pass	89.99	89.99	Testing that testOne has the correct price assigned.
Pass	'purple'	'purple'	Testing that testTwo has the correct color assigned.
Pass	25.0	25.0	Testing that testTwo has the correct color assigned.

You passed: 100.0% of the tests

Not yet graded

Create a class called `AppleBasket` whose constructor accepts two inputs: a string representing a color, and a number representing a quantity of apples. The constructor should initialize two instance variables: `apple_color` and `apple_quantity`. Write a class method called `increase` that increases the quantity by `1` each time it is invoked. You should also write a `__str__` method for this class that returns a string of the format: `"A basket of [quantity goes here] [color goes here] apples."` e.g. `"A basket of 4 red apples."` or `"A basket of 50 blue apples."` (Writing some test code that creates instances and assigns values to variables may help you solve this problem!)

Save & Run

5/15/2021, 1:08:21 PM - 2 of 2

Show in CodeLens

```
class AppleBasket:
    """ Point class for representing and manipulating x,y coordinates. """

    def __init__(self, initX, initY):

        self.apple_color = initX
        self.apple_quantity = initY

    def increase(self):
        return self.apple_quantity + 1

    def increase(self):
        self.apple_quantity = self.apple_quantity + 1

```

```
A basket of 4 red apples.
A basket of 5 red apples.
```

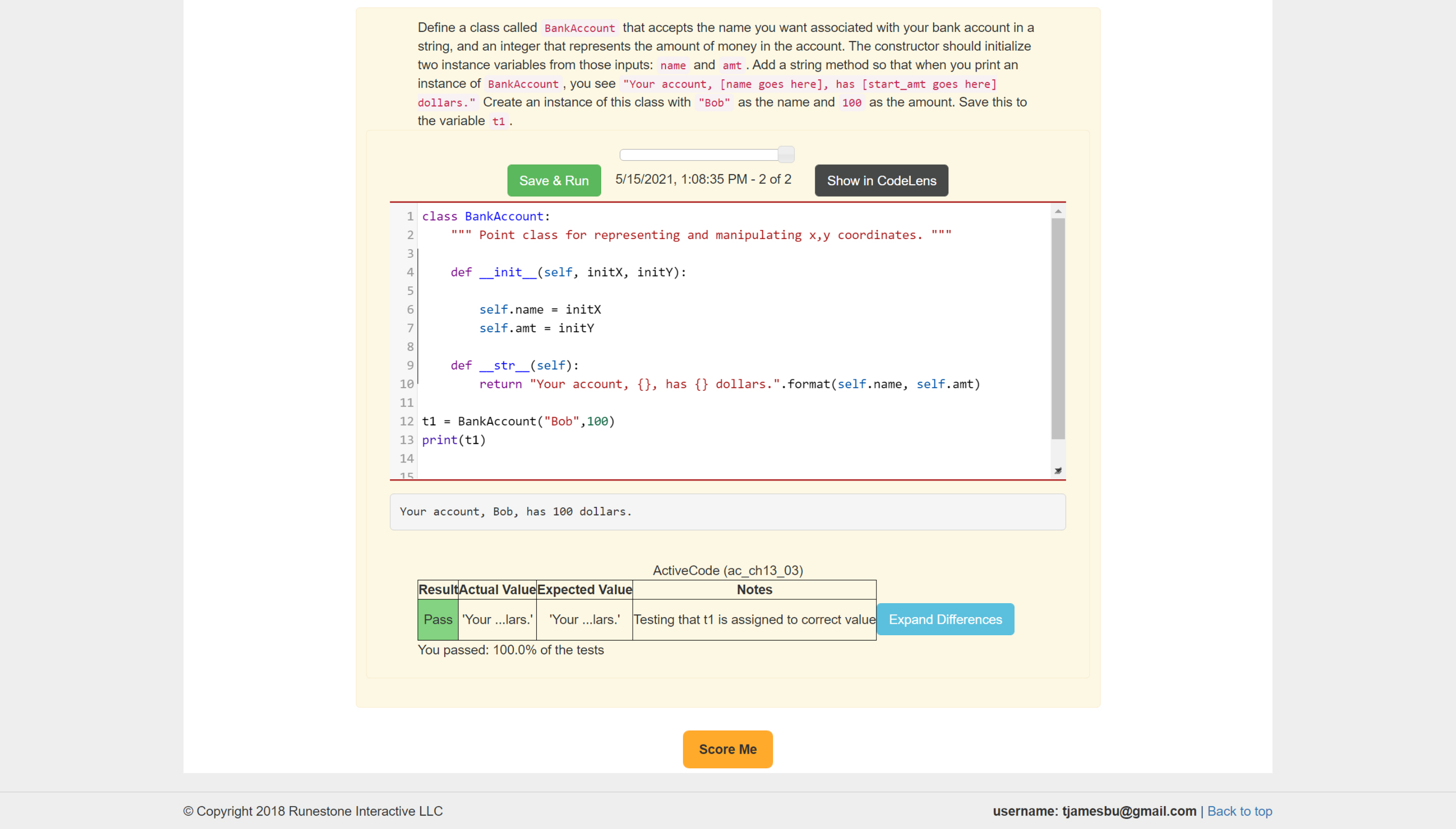
ActiveCode (ac_ch13_021)

Result	Actual Value	Expected Value	Notes
Pass	4	4	Testing the initialization of the apple_quantity inst var.
Pass	'A bas...ples.'	'A bas...ples.'	
Pass	5	5	Testing the increase method

Expand Differences

You passed: 100.0% of the tests

Not yet graded



Define a class called `BankAccount` that accepts the name you want associated with your bank account in a string, and an integer that represents the amount of money in the account. The constructor should initialize two instance variables from those inputs: `name` and `amt` . Add a string method so that when you print an instance of `BankAccount` , you see `"Your account, [name goes here], has [start_amt goes here] dollars."` Create an instance of this class with `"Bob"` as the name and `100` as the amount. Save this to the variable `t1` .

```
class BankAccount:
    """ Point class for representing and manipulating x,y coordinates. """

    def __init__(self, initX, initY):

        self.name = initX
        self.amt = initY

    def __str__(self):
        return "Your account, {}, has {} dollars.".format(self.name, self.amt)

t1 = BankAccount("Bob",100)
print(t1)

```

```
Your account, Bob, has 100 dollars.
```

ActiveCode (ac_ch13_03)

Result	Actual Value	Expected Value	Notes
Pass	'Your ...lars.'	'Your ...lars.'	Testing that t1 is assigned to correct value

You passed: 100.0% of the tests

username: tjamesbu@gmail.com | Back to top