Feedback — Quiz 7a

You submitted this quiz on **Sun 1 Dec 2013 7:04 AM PST**. You got a score of **100.00** out of **100.00**.

Question 1

Let's define a class for 2-dimensional points.

```
class Point2D:
    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y

    def translate(self, deltax = 0, deltay = 0):
        """Translate the point in the x direction by deltax
           and in the y direction by deltay."""
        self.x += deltax
        self.y += deltay

    …
```

Which of the following code snippets are valid usages of the `Point2D` initializer and its `translate` method? For your first attempt at this problem, we suggest that you try to answer without using CodeSkulptor.

Your Answer		Score	Explanation

☑ `point1 = Point2D(3, 9)`
`point2 = Point2D()`
`point2.translate(20, 4)`
	✔	6.00	Yes, you can define multiple `Point2D` objects. Furthermore, the initializer is defined so that you don't have to provide arguments to `Point2D()`.

☑ `point = Point2D(3, 9)` `point.tran`	✔	6.00	

slate(5, -2)			
☐ point = Point2D(3, 9) translate(point, 5, -2)	✔	1.50	No, `translate` is not defined globally. It is defined only for `Point2D` objects.
☐ Point2D = (3, 9) Point2D.translate(5, -2)	✔	1.50	
Total		15.00 / 15.00	

Question 2

Let's continue to use the same class for 2-dimensional points.

```
class Point2D:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

    def translate(self, deltax=0, deltay=0):
        """Translate the point in the x direction by deltax
           and in the y direction by deltay."""
        self.x += deltax
        self.y += deltay

    …
```

Which of the following code snippets are valid usages of the `Point2D` initializer and its `translate` method? For your first attempt at this problem, we suggest that you try to answer without using CodeSkulptor.

Your Answer	Score	Explanation

☐ points = [(2, 5), (8, 3), (0, 2)] for point in points: point.translate(-1, -1)	✔ 1.50	No, `translate` is defined only on a `Point2D` object, not on a tuple.
☐ point0 = Point2D(2, 5) point1 = Point2D(8, 3) point2 = Point2D(0, 2) points = [point0, point1, point2] points.translate(-1, -1)	✔ 1.50	No, `translate` is defined only on a `Point2D` object, not on a list of `Point2D` objects.
☑ point0 = Point2D(2, 5) point1 = Point2D(8, 3) point2 = Point2D(0, 2) points = [point0, point1, point2] for point in points: point.translate(-1, -1)	✔ 12.00	
Total	15.00 / 15.00	

Question 3

Let's continue to use the same class for 2-dimensional points.

```
class Point2D:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

    def translate(self, deltax=0, deltay=0):
        """Translate the point in the x direction by deltax
```

```
            and in the y direction by deltay."""
        self.x += deltax
        self.y += deltay

    …
```

Which of the following code snippets are valid usages of the `Point2D` initializer and its `translate` method? For your first attempt at this problem, we suggest that you try to answer without using CodeSkulptor.

Your Answer		Score	Explanation
☐ point = Point2D(3, 6) s = str(point) newpoint = Point(s)	✔	1.50	
☑ point = Point2D(3, 6) s = str(point)	✔	10.50	
☐ point = Point2D(3, 6) lst = list(point) x = lst[0]	✔	1.50	
☐ point = Point2D(3, 6) lst = list(point)	✔	1.50	
Total		15.00 / 15.00	

Question 4

In SimpleGUI, the function `draw_image` takes an optional sixth parameter that determines the angle of rotation of the destination rectangle around its center. Do positive values for the angle rotate the image clockwise or counterclockwise? Is the angle specified in degrees or radians?

Refer to the CodeSkulptor documentation.

Your Answer		Score	Explanation
○ counterclockwise, degrees			
◉ clockwise, radians	✔	10.00	

○ clockwise, degrees	
○ counterclockwise, radians	
Total	10.00 / 10.00

Question 5

One interesting extension of *Rice Rocks* would be to have two ships, with each controlled by a separate player, instead of one single ship. Using the provided class definitions, what is the best way to represent the two ships in this new variant?

Your Answer	Score	Explanation
○ In the `Ship` class definition, duplicate **every** method. For example, `Ship.draw1(…)` would be used to draw the first ship, while `Ship.draw2(…)` would be used to draw the second ship.		
◉ Add another call to the `Ship` constructor, assigning the result to another global variable. ship1 = Ship(…) ship2 = Ship(…)	✔ 15.00	

○ Copy the `Ship` class code, e.g.,
class Another_Ship:
  def __init__(self, pos, vel, angle):
  …
  …
Then create two ship objects, one of each class, assigning each to a global variable.
ship1 = Ship(…)
ship2 = Another_Ship(…)		

○ In the `Ship` class definition, change the variables `pos`, `vel`, `angle` to be lists of two values each. Then, change each method to take an additional number argument that indicates which ship should be used. Thus, when we call the		

constructor now, we are creating both ships.

ships = Ship(…)

Total	15.00 / 15.00

Question 6

Which of the following browsers fully support MP3 audio files? Refer to the CodeSkulptor documentation.

Your Answer		Score	Explanation
☑ Safari	✔	4.00	
☐ Firefox	✔	2.00	Firefox currently supports MP3 files on some, but not all systems.
☑ Chrome	✔	4.00	
Total		10.00 / 10.00	

Question 7

Consider a spaceship where the ship's thrusters can accelerate the ship by 10 pixels per second for each second that the thrust key is held down. If the friction induces a deceleration that is 10% of the ship's velocity per second, what is the maximal velocity of the ship? If you are having trouble, consider writing a short program to help understand this problem.

Your Answer	Score	Explanation
○ The ship has no maximal velocity. It can reach any velocity the player desires if		

you hold the thrust key down long enough.			
◉ Around 100 pixels per second	✔	20.00	At a velocity of 100 pixels per second, friction would induce a deceleration of 10 pixels per second. This deceleration would exactly cancel an acceleration of 10 pixels per second from the thrusters. We used “around” here since the true maximal velocity depends on the rate at which the frame is drawn.
○ Around 10 pixels per second			
○ Around 1000 pixels per second			
Total		20.00 / 20.00	