An Introduction To Interactive Programing In Python (Part 2)

by Joe Warren, Scott Rixner, John Greiner, Stephen Wong

Quiz 7a – More classes *Skanda S Bharadwaj*

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
point = Point2D(3, 9) translate(point, 5, -2)	Correct	1.50	No, `translate` is not defined globally. It is defined only for `Point2D` objects.
✓ point = Point 2D(3, 9) ✓ point.translate (5, -2)	Correct	6.00	
✓ point1 = Poin t2D(3, 9) ✓ point2 = Poin t2D() ✓ point2.transla te(20, 4)	Correct	6.00	Yes, you can define multiple `Point2D` objects. Furthermore, the initializer is defined so tha you don't have to provide arguments to `Point2D()`.
Point2D = (3, 9) Point2D.translate(5, -2)	Correct	1.50	

2

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)	Correct	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)	Inorrect	0.00	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, poi nt2] for point in points: point.translate(-1, -1)	Inorrect	0.00	

Total	1.50 / 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` class? Note that the three dots in the class definition indicates the possible existence of other class methods. 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)	Correct	10.50	
point = Point2D(3, 6) tup = tuple(point)	Correct	1.50	
point = Point2D(3, 6) s = str(point) newpoint = Point(s)	Correct	1.50	
point = Point2D(3, 6) lst = list(point) x = lst[0]	Correct	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
✓ clockwise, radians	Correct	10.00	
counterclockwise, radians			
clockwise, degrees			
counterclockwise, degrees			
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	Explanati

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.

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(…)
```

✓ Add another call to the `Ship` constructor, assigning the result to another global variable. Correct 15.00

```
ship1 = Ship(…)
ship2 = 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
Firefox	Correct	2.00	Firefox currently supports MP3 files on some, but not all systems.
✓ Chrome	Correct	4.00	
✓ Safari	Correct	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

✓ Around 100 pixels per second	Correct	20.00	At a velocity of 100 pixels per second, friction would induce a deceleration 10 pixels per second. This deceleration would exactly cancel an acceleratio 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.
The ship has no maximal velocity. It can reach any velocity the player desires if you hold the thrust key down long enough.			
Around 1000 pixels per second			
Around 10 pixels per second			
Total		20.00 / 20.00	