1. In Processing with Python mode, loading an image into your sketch is a common operation when working with multimedia applications. This allows you to display, manipulate, and analyze images within your sketches. What is the correct way to load an image file named "landscape.jpg" located in the sketch's "data" folder, so it can be displayed within your Processing sketch?

*def setup():*

*global img*

*img = loadImage("landscape.jpg")*

Question 2

In Processing using Python mode, organizing project files is crucial for maintaining a clean workspace, especially when working with external media like images. Suppose you have a folder named "images" within your sketch folder that contains all the image files you intend to use in your sketch. How would you correctly specify the path to load an image named "sunset.jpg" from this "images" folder?

*def setup():*

*global img*

*img = loadImage("images/sunset.jpg")*

Question 3

In Processing with Python mode, the scale() function can be used to apply various transformations to images, including scaling their size up or down or even mirroring them. If you want to flip an image horizontally, creating a mirror image effect, which of the following code snippets correctly utilizes the scale() function to achieve this horizontal flip?

*def setup():*

*global img*

*img = loadImage("portrait.jpg")*

*imageMode(CENTER)*

*def draw():*

*background(240)*

*translate(width / 2, height / 2)*

*scale(-1.0, 1.0)*

*image(img, 0, 0)*

Question 5

In Processing using Python mode, you are tasked with creating a visually striking grid of rectangles. This grid should be 10 by 10, with each rectangle having a width and height of 40 pixels. The unique twist is that each row of rectangles should have an incremental rotation applied to it, starting with the first row having no rotation (0 degrees) and each subsequent row rotating an additional 4.5 degrees more than the previous one, culminating in the last row being rotated by 45 degrees. How can you implement this pattern using only for loops and transformations, ensuring that each row's rotation incrementally increases?

*rectMode(CENTER)*

*for i in range(10):*

*for j in range(10):*

*pushMatrix()*

*translate(j \* 80 + 40, i \* 80 + 40)*

*rotate(radians(4.5 \* i))*

*rect(0, 0, 40, 40)*

*popMatrix()*

Question 6

In Processing using Python mode, you're tasked with creating an engaging 5x5 grid of images on the canvas. For each position in the grid, there should be a 50% chance that the image displayed is either image1.jpg or image2.jpg, both of which are loaded at the start of the program. How would you implement this grid to ensure that each image has an equal chance of being one of the two options?

*def setup():*

*global img1, img2*

*size(500, 500)*

*img1 = loadImage("image1.jpg")*

*img2 = loadImage("image2.jpg")*

*noLoop()*

*def draw():*

*for i in range(5):*

*for j in range(5):*

*if random(1) < 0.5:*

*image(img1, j \* 100, i \* 100, 100, 100)*

*else:*

*image(img2, j \* 100, i \* 100, 100, 100)*

Question 7

In Processing with Python mode, you want to display an image named "landscape.jpg" at the center of the canvas. The goal is to rotate this image by 30 degrees and scale it to 25% of its original size. Which of the following code snippets correctly implements this transformation setup, ensuring the image is correctly positioned, rotated, and scaled as described?

*def setup():*

*global img*

*size(600, 400)*

*img = loadImage("landscape.jpg")*

*noLoop()*

*def draw():*

*background(240)*

*translate(width / 2, height / 2)*

*rotate(radians(30))*

*scale(0.25)*

*imageMode(CENTER)*

*image(img, 0, 0)*

Question 8

In Processing with Python mode, you are challenged to create a visually captivating sequence where multiple instances of the same image rotate around the center of the canvas, creating a full circle from 0 to 360 degrees. Each image in this sequence should have a 30% transparency to allow for overlapping effects. How would you implement this using a for loop, appropriate transformations, and transparency settings to achieve the effect of the images making a complete rotation with overlapping transparency?

*def setup():*

*global img*

*size(600, 600)*

*img = loadImage("flower.jpg")*

*noLoop()*

*def setup():*

*global img*

*size(600, 600)*

*img = loadImage("flower.jpg")*

*noLoop()*

*def draw():*

*background(240)*

*translate(width / 2, height / 2)*

*for angle in range(0, 360, 10):*

*pushMatrix()*

*rotate(radians(angle))*

*tint(255, 30)*

*imageMode(CENTER)*

*image(img, 0, 0)*

*popMatrix()*