Question 1

Conditional statements are fundamental in programming, allowing code to execute differently based on varying conditions or inputs. In Python, if, elif, and else statements enable such decision-making capabilities. Given a scenario where you need to print messages based on a temperature reading (an integer value stored in the variable temperature), which of the following code snippets correctly implements a series of conditional statements to print whether it's cold (temperature below 15), warm (temperature between 15 and 25, inclusive), or hot (temperature above 25)?

*if temperature < 15:*

*print("It's cold.")*

*elif temperature <= 25:*

*print("It's warm.")*

*else:*

*print("It's hot.")*

2. In Processing using Python mode, you want to create an interactive sketch that dynamically responds to the mouse's position on the canvas. The goal is to draw an ellipse when the mouse is on the left half of the screen and a rectangle when the mouse is on the right half of the screen. Considering the canvas's width and the mouse's x-coordinate (mouseX), which of the following code snippets correctly implements this behavior?

*if mouseX < width / 2:*

*ellipse(mouseX, mouseY, 50, 50)*

*else:*

*rect(mouseX - 25, mouseY - 25, 50, 50)*

Question 3

In Processing using Python mode, you aim to fill the canvas with a grid of circles. To add visual interest, approximately 20% of these circles should be randomly colored red, with the remainder colored white. Utilizing only for loops and if statements, and assuming the random selection process to determine each circle's color, which of the following code snippets correctly implements this design, adhering to the requirement that about 20% of the circles are red?

*rows = 10*

*cols = 10*

*spacing = width / cols*

*for y in range(rows):*

*for x in range(cols):*

*if random(1) < 0.2:*

*fill(255, 0, 0) # Red*

*else:*

*fill(255) # White*

*ellipse(x \* spacing + spacing / 2, y \* spacing + spacing / 2, spacing \* 0.8, spacing \* 0.8)*

Question 4

In Processing with Python mode, you are tasked with creating a visually compelling effect on a canvas: 100 rectangles span horizontally across the canvas. Each rectangle's fill color should be determined randomly, with the probability of a rectangle being filled with white starting at 100% on the left side of the canvas and gradually decreasing to 0% on the right side, effectively transitioning the likelihood of white to black across the canvas. Using only for loops and if statements, and considering a linear gradient of probability for the color fill of these rectangles, which of the following code snippets correctly implements this dynamic color transition?

*for i in range(100):*

*x = i \* (width / 100)*

*if random(1) < (1 - i / 100.0):*

*fill(255) # White*

*else:*

*fill(0) # Black*

*rect(x, 0, width / 100, height)*

Question 5

In Python, tuples are versatile data structures that can contain elements of different data types, including integers, floats, and strings. Given this capability, consider a tuple named my\_info that contains an integer representing your age, a float representing your height in meters, and a string with your first name. How would you correctly create this tuple and print a message that includes the string from the tuple?

*my\_info = (30, 1.75, "Alex")*

*print("Hello, my name is " + my\_info[2] + ".")*

6. In Python, tuples are immutable sequences that can store a variety of data types, including other tuples. This feature makes them particularly useful for grouping related data together, such as RGB color values for graphical applications. Given this capability, how would you correctly create a tuple named colors that contains three nested tuples, each representing a color in RGB format?

*colors = ((255, 0, 0), (0, 255, 0), (0, 0, 255))*

Question 7

In Processing using Python mode, state machines are useful for controlling the flow and behavior of sketches based on states and inputs. Imagine a simple state machine that controls the color of the canvas background: it switches between red and blue states when the mouse is clicked. Among the following code snippets, which one correctly implements this state machine, utilizing mouse clicks to transition between states?

*state = 'RED'*

*def setup():*

*size(400, 400)*

*def draw():*

*global state*

*if state == 'RED':*

*background(255, 0, 0)*

*elif state == 'BLUE':*

*background(0, 0, 255)*

*def mouseClicked():*

*global state*

*if state == 'RED':*

*state = 'BLUE'*

*else:*

*state = 'RED'*

Question 8

In Processing using Python mode, consider a simple state machine that controls the canvas's background color. Initially, the canvas is red. Upon each mouse click, there is a 30% chance that the state will change to blue. If the state is already blue, a mouse click will change it back to red, also with a 30% chance. Among the following code snippets, which one correctly implements this probabilistic state transition mechanism?

*state = 'RED'*

*def setup():*

*size(400, 400)*

*def draw():*

*global state*

*if state == 'RED':*

*background(255, 0, 0)*

*else:*

*background(0, 0, 255)*

*def mouseClicked():*

*global state*

*if random(1) < 0.3:*

*if state == 'RED':*

*state = 'BLUE'*

*else:*

*state = 'RED'*