

Problem Decomposition:

- The problem of creating a grid of nested, randomly colored circles was broken down into smaller tasks:
 - Set up a 5x5 grid layout.
 - Create a function to draw individual circles with random colors and sizes.
 - Use nested loops to handle positioning and rendering of each circle cluster.

Pattern Recognition:

- The solution recognizes a repeating pattern where each "tile" (or grid cell) contains a nested sequence of circles.
- This pattern allows for easy looping: a 5x5 outer grid and an inner loop to create 10 circles within each tile.

Abstraction:

- Functions like `drawCircles()` help encapsulate the drawing of nested circles, making it easier to reuse and manage.
- Using variables like `sz` for size and `gap` for spacing generalizes the code so that grid and circle properties are easily adjustable without modifying core logic.

Algorithmic Thinking:

- **Grid Creation:** Using two for loops to iterate over rows and columns sets up the grid efficiently.
- **Randomized Circle Creation:** Inside each grid cell, another for loop generates 10 nested circles. By applying a probability factor (`chanceFac < 0.9`), each circle has a 90% chance of appearing, adding randomness and variety.
- **Color Randomization:** Using `fill(random(255), random(255), random(255))` before each circle is drawn ensures each circle has a unique color. This randomness gives each grid tile a visually distinct look.