```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;

import javax.swing.JFrame;
import javax.swing.JPanel;

public class Frame extends JPanel{

    public void paint(Graphics pen) {
        super.paintComponent(pen); //default code, don't alter
        // Add code below
        pen.setColor(Color.black); //change pen color to black
        pen.fillRect(0, 0, 800, 600); //fill the background with black canvas

        //change pen color to white
        pen.setColor(Color.white);
        pen.drawRect(10, 10, 780, 540);
        pen.drawLine(780/2+10, 10, 780/2+10, 550); //vertical line to split rectangle
        pen.drawLine(10, 540/2+10, 790, 540/2+10); //horizontal line to split rectangle

        //start Student solution below

        //top-left quadrant (Lines)
        int cntr = 0;
        while (cntr < 50) {
            // generate a random x1 value between 10 and 400
            // Random Number Generation Formula: (int)(Math.random()*range) + min where range is max-min+1
            int x1 = (int)(Math.random()*(390)) + 10; // min x is 10, max is 400 for top-left quadrant
            int y1 = (int)(Math.random()*(270)) + 10; // min y is 10, max is 280 for top-left quadrant
            int x2 = (int)(Math.random()*(390)) + 10; // generate another set of random numbers for the 2nd x value
            int y2 = (int)(Math.random()*(270)) + 10;

            // draw the line using the generated random values for the end points of the line
            int r = (int)(Math.random()* 256);
            int g = (int)(Math.random()* 256);
            int b = (int)(Math.random()* 256);

            // change color of pen
            pen.setColor(new Color(r, g, b));

            pen.drawLine(x1, y1, x2, y2);

            cntr++; // update loop counter
        }

        //bottom-left quadrant (Circles)
        cntr = 0;
        while (cntr < 50) {
            // Randomly generated circle sizes between 5 - 100
            int size = (int)(Math.random()* 200) + 5;

            // Randomly generated x1 & y1 values so circles don't break quadrant boundaries
            int x1 = (int)(Math.random()*(380 - size)) + 10; // seems to be the only way to make this work
            int y1 = (int)(Math.random()*(260 - size)) + 290; // seems to be the only way to make this work

            // draw the line using the generated random values for the end points of the line
            int r = (int)(Math.random()* 256);
            int g = (int)(Math.random()* 256);
            int b = (int)(Math.random()* 256);

            // change color of pen
            pen.setColor(new Color(r, g, b));
            // draw circles
            pen.drawOval(x1, y1, size, size);

            cntr++; // Update loop counter
        }

        //top-right quadrant (Squares)
        cntr = 0;
        while (cntr < 50) {
            // Set size of square to 25 pixels
            int size = 50;

            // Randomly generated x1 & y1 values
            int x1 = (int)(Math.random()*(380 - size + 1)) + 402;
            int y1 = (int)(Math.random()*(250 - size + 2)) + 15;

            // draw the line using the generated random values for the end points of the line
            int r = (int)(Math.random()* 256);
            int g = (int)(Math.random()* 256);
            int b = (int)(Math.random()* 256);

            // change color of pen
            pen.setColor(new Color(r, g, b));
            // draw Squares
            pen.fillRect(x1, y1, size, size);

            cntr++; // Update loop counter

        }

        //bottom-right quadrant (3-D Cube with 4 colors)
        cntr = 0;

        for (int i = 0; i < 4; i++) {

            for(int x = 582 ; x < 631; x++) {
                for(int y = 330; y < 400; y++) {
                    pen.setColor(Color.blue);
                    pen.drawLine(x, y, x + 49, y + 50);
                }
                for(int x1 = 530 ; x1< 630  ; x1++) {
                    for(int y1 = 330 ; y1 < 430; y1++) {
                        pen.setColor(Color.yellow);
                        pen.drawLine(x1, y1, x1, y1);
                    }
                }
                for(int x2 = 530; x2 < 531 ; x2++) {
                    for(int y2 = 330 ; y2 < 430 ; y2++) {
                       pen.setColor(Color.green);
                       pen.drawLine(x2, y2, x2 + 50, y2 + 50);
                    }
                 }
                 for(int x3 = 580; x3 < 680; x3++) {
                    for(int y3 = 380; y3 < 480; y3++) {
                       pen.setColor(Color.red);
                       pen.drawLine(x3, y3, x3, y3);
                    }
                 }
              }
              cntr++;
              System.out.println(cntr);
           }


        } //end of paint - do not accidentally delete

        public static void main(String[] arg) {
           Frame f = new Frame();
        }//end of main - do not delete

        public Frame() {
           JFrame f = new JFrame("Loops and Random");
           f.setSize(new Dimension(800,600));
           f.add(this);
           f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        f.setVisible(true);
    } //end of constructor - do not delete


} //end of class - do not delete
```