

```

import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.awt.Color;
import java.util.Random;

public class LineArt extends JPanel {

    // Unique version ID for this class to ensure saved objects can be loaded safely
    private static final long serialVersionUID = 1L;

    // Initial width of height of the starting rectangle
    private static int width = 980;
    private static int height = 630;

    // main method to launch the program as a standalone application - no need to
    // modify
    public static void main(String[] args) {
        LineArt panel = new LineArt();
        panel.setPreferredSize(new Dimension(width + 20, height + 20)); // content size
        window dimensions

        JFrame frame = new JFrame("Line Art"); // Title of frame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(panel);
        frame.pack();
        frame.setVisible(true);
    }

    /**
     * Draw the four corners of line art. Line art displays straight lines inside a rectangle from
     one side to
     * a perpendicular side. The lines must be drawn in such a way that both the starting
     points of the lines on
     * one side and the ending points on the other side are equi-distant along the sides. The
     size of the rectangle
     * is 980 pixels wide by 630 pixels high.
     *
     * @param g the Graphics object used for drawing shapes, text, and images
     */
    public void drawLineArt(Graphics g) {

        // Draw the initial rectangle

```

```

g.drawRect(10, 10, width, height);
int y = 0;

// Draw bottom-left corner

for (int j = 10; j<=width; j+=16) {
    g.setColor(RandomColor());
    g.drawLine(10, y=y+10, j, 640); //10,640 is corner
}

// Draw bottom-right corner
int x=0;

for (int j = 990; j>=0; j-=16) {
    g.setColor(RandomColor());
    g.drawLine(990, x=x+10, j, 640);
}

// Draw top-right corner
int q = 640;

for (int j = 990; j>=0; j-=16) {
    g.setColor(RandomColor());
    g.drawLine(990, q=q-10, j, 10);
}

// Draw top-left corner

int a = 640;

for (int j = 10; j<=width; j+=16) {
    g.setColor(RandomColor());
    g.drawLine(10, a=a-10, j, 10);
}

}

/**
 * Overrides JPanel's paintComponent method to perform custom drawing.
 *
 * @param g the Graphics object used for drawing shapes, text, and images
 */
@Override
protected void paintComponent(Graphics g) {

```

```
        super.paintComponent(g); // Clears the panel before drawing
        drawLineArt(g);
    }

    private static Color RandomColor() {
        Random rand = new Random();
        int r = rand.nextInt(256);
        int green = rand.nextInt(256);
        int b = rand.nextInt(256);

        Color randomColor = new Color(r, green, b);
        return randomColor;
    }
}
```