```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
import javax.swing.JFrame;
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) {
        Random rand = new Random();
        // Draw the initial rectangle
        g.drawRect(10, 10, width, height);
        int x1 = 0;
        int y1 = 0;
        int x2 = 0;
        int y2 = 630;
        int red = 0;
        int green = 0;
        int blue = 0;
        int i = 9;
        // Draw bottom-left corner
        while (y1<=630&&x2<=980) {
                red = rand.nextInt(256);
                green = rand.nextInt(256);
                blue = rand.nextInt(256);
                Color color = new Color(red,green,blue);
                g.setColor(color);
                g.drawLine(x1+10,y1+10,x2+10,y2+10);
                y1= y1+i;
                x2 = x2+i;
        }

        // Draw bottom-right corner
        x1=980;
        y1=0;
        x2=980;
        y2=630;
        while (y1<=630&&x2>=10) {
                red = rand.nextInt(256);
                green = rand.nextInt(256);
                blue = rand.nextInt(256);
                Color color = new Color(red,green,blue);
                g.setColor(color);
                g.drawLine(x1+10,y1+10,x2+10,y2+10);
                y1= y1+i;
                x2 = x2-i;
        }
```

```
// Draw top-right corner
x1=980;
y1=0;
x2=980;
y2=630;
while (x1>=10 &&y2>=10) {
        red = rand.nextInt(256);
        green = rand.nextInt(256);
        blue = rand.nextInt(256);
        Color color = new Color(red,green,blue);
        g.setColor(color);
        g.drawLine(x1+10,y1+10,x2+10,y2+10);
        x1= x1-i;
        y2 = y2-i;
}
// Draw top-left corner
x1=0;
y1=0;
x2=0;
y2=630;
while (x1<=980 &&y2>=10) {
        red = rand.nextInt(256);
        green = rand.nextInt(256);
        blue = rand.nextInt(256);
        Color color = new Color(red,green,blue);
        g.setColor(color);
        g.drawLine(x1+10,y1+10,x2+10,y2+10);
        x1= x1+i;
        y2 = y2-i;
}

x1 = 235;
y1 = 115;
x2 = 235;
y2 = 535;
// Draw bottom-left corner
while (y1<=535&&x2<=765) {
        red = rand.nextInt(256);
        green = rand.nextInt(256);
        blue = rand.nextInt(256);
```

```
            Color color = new Color(red,green,blue);
            g.setColor(color);
            g.drawLine(x1,y1,x2,y2);
            y1= y1+i;
            x2 = x2+i;
}
x1 = 765;
y1 = 115;
x2 = 765;
y2 = 535;
// Draw bottom-right corner
while (y1<=535&&x2>=235) {
            red = rand.nextInt(256);
            green = rand.nextInt(256);
            blue = rand.nextInt(256);
            Color color = new Color(red,green,blue);
            g.setColor(color);
            g.drawLine(x1,y1,x2,y2);
            y1= y1+i;
            x2 = x2-i;
}
x1 = 765;
y1 = 115;
x2 = 765;
y2 = 535;
// Draw top-right corner
while (y2>=115&&x1>=235) {
            red = rand.nextInt(256);
            green = rand.nextInt(256);
            blue = rand.nextInt(256);
            Color color = new Color(red,green,blue);
            g.setColor(color);
            g.drawLine(x1,y1,x2,y2);
            x1= x1-i;
            y2 = y2-i;
}
x1 = 235;
y1 = 115;
x2 = 235;
y2 = 535;
```

```
            // Draw top-right corner
            while (y2>=115&&x1<=765) {
                  red = rand.nextInt(256);
                  green = rand.nextInt(256);
                  blue = rand.nextInt(256);
                  Color color = new Color(red,green,blue);
                  g.setColor(color);
                  g.drawLine(x1,y1,x2,y2);
                  x1= x1+i;
                  y2 = y2-i;
            }


      }

      /**
       * 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);
      }

}
```