```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Graphics;
import java.util.Random;
import javax.swing.JPanel;
import javax.swing.JFrame;
public class LineArt extends JPanel {
	Random randy = new Random();
	// 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);
		// Draw bottom-left corner
```

```
int y = 10;
for(int x = 10; x<=990; x+=30)
{
	int red = randy.nextInt(256);
	int blue = randy.nextInt(256);
	int green = randy.nextInt(256);
	Color set = new Color(red, blue, green);

	g.setColor(set);
	g.drawLine(x, 640, 10, y);
	y = y + 30*65/100;
}
// Draw bottom-right corner

y=640;
for(int x = 990; x>=10; x-=30)
{
	int red = randy.nextInt(256);
	int blue = randy.nextInt(256);
	int green = randy.nextInt(256);
	Color set = new Color(red, blue, green);

	g.setColor(set);
	g.drawLine(x, 10, 990, y);
	y = y - 30*65/100;
}
// Draw top-right corner
y = 10;
for(int x = 990; x>=10; x-=30)
{
	int red = randy.nextInt(256);
	int blue = randy.nextInt(256);
	int green = randy.nextInt(256);
	Color set = new Color(red, blue, green);

	g.setColor(set);
	g.drawLine(x, 640, 990, y);
	y = y + 30*65/100;
}
// Draw top-left corner
```

```
y = 640;
for(int x = 10; x<=990; x+=30)
{
        int red = randy.nextInt(256);
        int blue = randy.nextInt(256);
        int green = randy.nextInt(256);
        Color set = new Color(red, blue, green);

        g.setColor(set);
        g.drawLine(x, 10, 10, y);
        y = y - 30*65/100;
}


//Rectangle inside the main rectangle
int s = 480;
g.drawRect(500-s/2, 325-s/2*65/100, s, s*65/100);

y = 481;
for(int x = 260; x<=740; x+=30)
{
        int red = randy.nextInt(256);
        int blue = randy.nextInt(256);
        int green = randy.nextInt(256);
        Color set = new Color(red, blue, green);

        g.setColor(set);
        g.drawLine(x, 481, 740, y);
        y = y - 30*65/100;
}
//Draw bottom-right corner
y = 481;
for(int x = 740; x>=260; x-=30)
{
        int red = randy.nextInt(256);
        int blue = randy.nextInt(256);
        int green = randy.nextInt(256);
        Color set = new Color(red, blue, green);

        g.setColor(set);
```

```
                g.drawLine(260, y, x, 481);
                y = y - 30*65/100;
        }
        y = 169;
        for(int x = 740; x>=260; x-=30)
        {
                int red = randy.nextInt(256);
                int blue = randy.nextInt(256);
                int green = randy.nextInt(256);
                Color set = new Color(red, blue, green);

                g.setColor(set);
                g.drawLine(260, y, x, 169);
                y = y + 30*65/100;
        }
        y = 169;
        for(int x = 260; x<=740; x+=30)
        {
                int red = randy.nextInt(256);
                int blue = randy.nextInt(256);
                int green = randy.nextInt(256);
                Color set = new Color(red, blue, green);

                g.setColor(set);
                g.drawLine(x, 169, 740, y);
                y = y + 30*65/100;
        }

}
/**
 * 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);
}
```

```
}
```