```
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
import javax.swing.JFrame;

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);
```

```
// Draw bottom-left corner

int BLy = 630;

for(int BLx = 890; BLx > 0; BLx -= 10) {

        g.drawLine(BLx, height + 10, 10, BLy);

        BLy -= 7;

}

// Draw bottom-right corner
int BRy = 630;

for(int BRx = 110; BRx < 990; BRx += 10) {

        g.drawLine(BRx, height + 10, width + 10, BRy);

        BRy -= 7;

}

// Draw top-right corner

int TRy = 10;

for(int TRx = 110; TRx < 990; TRx += 10) {

        g.drawLine(TRx + 10, 10, width + 10, TRy);

        TRy += 7;

}

// Draw top-left corner

int TLy = 0;

for(int TLx = 890; TLx > 0; TLx -= 10) {

        g.drawLine(TLx, 10, 10, TLy + 10);

        TLy += 7;
```

```
        }

        // Draw the new initial rectangle



        int midX = 10 + width / 2;   // 10 + 490 = 500
        int midY = 10 + height / 2;  // 10 + 315 = 325


        int innerWidth =  (30 + width) / 2; //(980 + 30) / 2 = 505
        int innerHeight = (30 + height) / 2; //(630 + 30) / 2 = 330

        int innerX = midX - innerWidth / 2; //247
        int innerY = midY - innerHeight / 2; //160




// Recursively draw the same pattern inside the smaller rectangle
        g.drawRect(innerX, innerY, innerWidth, innerHeight);



        // Draw bottom-left corner (248, 490)

        int BBLy = 490;

        for(int BBLx = 700; BBLx > 248; BBLx -= 10) {

                g.drawLine(BBLx, 490, 248, BBLy);

                BBLy -= 7;

        }

        // Draw bottom-right corner (752, 490)
        int BBRy = 490;

        for(int BBRx = 300; BBRx < 752; BBRx += 10) {

                g.drawLine(BBRx, 490, 752, BBRy);
```

```
                BBRy -= 7;

                }

        // Draw top-right corner (752, 160)
        int TTRy = 160;

        for(int TTRx = 300; TTRx < 752; TTRx += 10) {

                g.drawLine(TTRx, 160, 752, TTRy);

                TTRy += 7;

        }

        // Draw top-left corner (248, 160)

        int TTLy = 160;

        for(int TTLx = 700; TTLx > 248; TTLx -= 10) {

                g.drawLine(TTLx, 160, 248, TTLy);

                TTLy += 7;

        }


    }



    /**
     * 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);
    }

}
```