```
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 lines = 50;
double xspace = width / (lines - 1.0);
double yspace = height / (lines - 1.0);

for ( int i = 0; i < lines; i++) {
        int x1 = 10;
        int y1 = 10 + (int) (yspace * i);
        int x2 = 10 + (int) (xspace * i);
        int y2 = height + 10;
        g.drawLine(x1, y1, x2, y2);
}

// Draw bottom-right corner

for ( int i = 0; i < lines; i++) {
        int x1 = width + 10;
        int y1 = 10 + (int) (yspace * i);
        int x2 = width + 10 - (int) (xspace * i);
        int y2 = height + 10;
        g.drawLine(x1, y1, x2, y2);
}

// Draw top-left corner
for (int i = 0; i < lines; i++) {
        int x1 = 10;
        int y1 = height + 10 - (int)(yspace * i);
        int x2 = 10 + (int)(xspace * i);
        int y2 = 10;
        g.drawLine(x1, y1, x2, y2);
}


// Draw top-right corner
for (int i = 0; i < lines; i++) {
        int x1 = width + 10;
        int y1 = height + 10 - (int)(yspace * i);
        int x2 = width + 10 - (int)(xspace * i);
        int y2 = 10;
        g.drawLine(x1, y1, x2, y2);
}

// drawing little rect
```

```
int littleX = 10 + width / 4;
int littleY = 10 + height / 4;
int littleH = (int)(height * 0.5);
int littleW = (int)(width * 0.5);

g.drawRect(littleX, littleY, (int) (width * 0.5), (int) (height * 0.5));

// drawing little drawing
lines = 25;
xspace = width * 0.5 / (lines - 1.0);
yspace = height * 0.5 / (lines - 1.0);
for ( int i = 0; i < lines; i++) {
        int x1 = littleX;
        int y1 = littleY + (int) (yspace * i);
        int x2 = littleX + (int) (xspace * i);
        int y2 = littleY + littleH;
        g.drawLine(x1, y1, x2, y2);
}

for ( int i = 0; i < lines; i++) {
        int x1 = littleX + littleW;
        int y1 = littleY + (int) (yspace * i);
        int x2 = littleW + littleX - (int) (xspace * i);
        int y2 = littleH + littleY;
        g.drawLine(x1, y1, x2, y2);
}

for (int i = 0; i < lines; i++) {
        int x1 = littleX;
        int y1 = littleH + littleY - (int)(yspace * i);
        int x2 = littleX + (int)(xspace * i);
        int y2 = littleY;
        g.drawLine(x1, y1, x2, y2);
}


for (int i = 0; i < lines; i++) {
        int x1 = littleW + littleX;
        int y1 = littleH + littleY - (int)(yspace * i);
        int x2 = littleW + littleX - (int)(xspace * i);
        int y2 = littleY;
        g.drawLine(x1, y1, x2, y2);

}
```

```
        }

        /**
         * 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);
        }

}
```