```
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 = 10;
		for (int x = 10; x < 990; x = x + 14) {
			g.drawLine(x, 640, 10, bly);
			bly = bly + 9;
		}
		// Draw bottom-right corner
		int bry = 10;
```

```
            for (int x = 990; x > 10; x = x - 14) {
                  g.drawLine(x, 640, 990, bry);
                  bry = bry + 9;
            }

            // Draw top-right corner
            int ty = 640;
            for (int x = 990; x > 10; x = x - 14) {
                  g.drawLine(x, 10, 990, ty);
                  ty = ty - 9;
            }
            // Draw top-left corner
            int tly = 640;
            for (int x = 10; x < 990; x = x + 14) {
                  g.drawLine(x, 10, 10, tly);
                  tly = tly - 9;
            }

            //Draw inner box
            g.drawRect(255, 167, 490, 315);

            //Draw inner b-l corner
            int ibly = 167;
            for (int x = 255; x < 745; x = x + 8) {
                  g.drawLine(x, 482, 255, ibly);
                  ibly = ibly + 5;
            }

            // Draw inner b-r corner
            int ibry = 167;
            for (int x = 745; x > 255; x = x - 8) {
                  g.drawLine(x, 482, 745, ibry);
                  ibry = ibry + 5;
            }

            // Draw inner t-r corner
            int ity = 482;
            for (int x = 745; x > 255; x = x - 8) {
                  g.drawLine(x, 167, 745, ity);
                  ity = ity - 5;
            }

            // Draw inner t-l corner
            int itly = 482;
            for (int x = 255; x < 745; x = x + 8) {
                  g.drawLine(x, 167, 255, itly);
                  itly = itly - 5;
            }

      }

      /**
       * 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);
      }

}
```