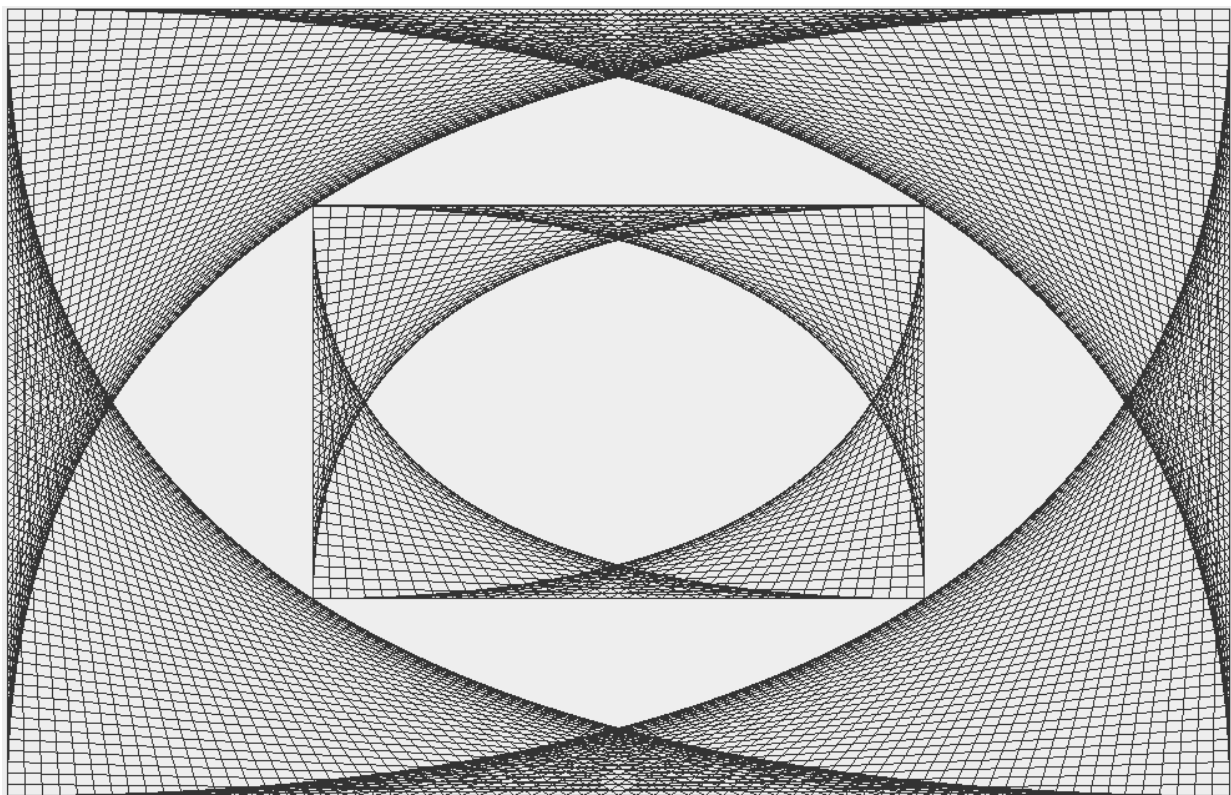




```

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);
        int x = 10;
        int y = 10;

        // Draw bottom-left corner

        for (x = 10; x < width + 10; x = x + 14) {

```

```

        g.drawLine(10, y, x, height + 10);
        y = y + 9;
    }
    y = 10;

    // Draw top-left corner
    for (x = 990; x > 10; x = x - 14) {
        g.drawLine(x, 10, 10, y);
        y = y + 9;
    }
    y = 10;

    // Draw top-right corner

    for (x = 10; x < width+10; x = x + 14) {
        g.drawLine(x, 10, width + 10, y);
        y = y + 9;
    }
    y = 640;

    // Draw bottom-right corner

    for (x = 10; x < width+10; x = x + 14) {
        g.drawLine(x, 640, width + 10, y);
        y = y - 9;
    }

    g.drawRect((width / 4) + 10, (height / 4) + 10, width / 2,
height / 2);

    y=168;

    // Draw bottom-left corner

    for (x = 255; x < 745; x = x + 14) {
        g.drawLine(255, y, x, 482);
        y = y + 9;
    }
    y = 168;

    // Draw top-left corner
    for (x = 745; x > 255; x = x - 14) {
        g.drawLine(x, 168, 255, y);
        y = y + 9;
    }
    y = 168;

    // Draw top-right corner

    for (x = 255; x < 745; x = x + 14) {
        g.drawLine(x, 168, 745, y);
        y = y + 9;
    }
    y = 482;

```

```

        // Draw bottom-right corner

        for (x = 255; x < 745; x = x + 14) {
            g.drawLine(x, 482, 745, y);
            y = y - 9;
        }
    }

    /**
     * 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);
    }
}

```