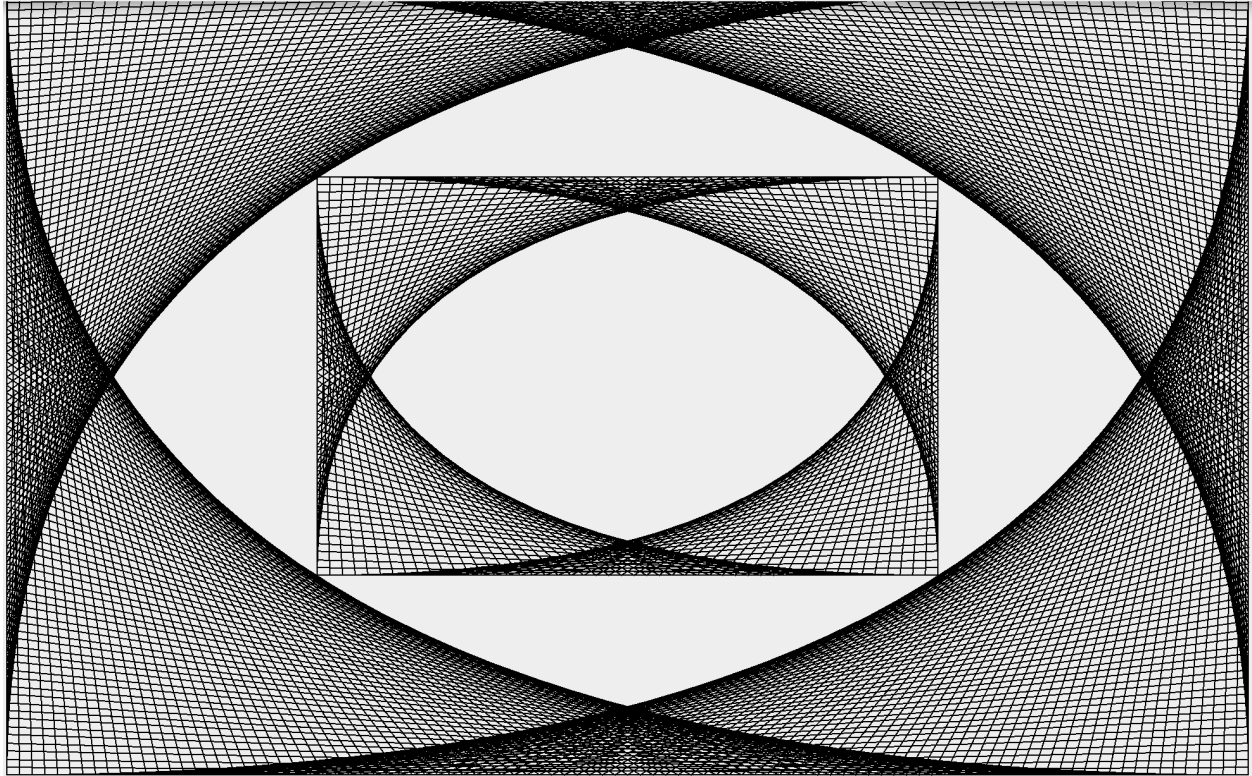




```
import java.awt.Dimension;  
import java.awt.Graphics;  
import javax.swing.JPanel;  
import javax.swing.JFrame;
```

```
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
    for (int i=0; i<=980; i=i+10) {
        g.drawLine(990-i,640,10,640-(int)(i*(((double)height/width))));
    }

    // Draw bottom-right corner
    for (int i=0; i<=980; i=i+10) {
        g.drawLine(i+10,640,990,640-(int)(i*(((double)height/width))));
    }

    // Draw top-right corner
    for (int i=0; i<=980; i=i+10) {
        g.drawLine(i+10,10,990,10+(int)(i*(((double)height/width))));
    }

    // Draw top-left corner
    for (int i=0; i<=980; i=i+10) {
        g.drawLine(990-i,10,10,10+(int)(i*(((double)height/width))));
    }
}

```

```

    }

    // Draw the inner rectangle
    g.drawRect(255, 167, width/2, height/2);
    // Draw bottom-left corner
    for (int i=0; i<=490; i=i+10) {
        g.drawLine(745-i,482,255,482-(int)(i*(((double)height/width))));
    }

    // Draw bottom-right corner
    for (int i=0; i<=490; i=i+10) {
        g.drawLine(i+255,482,745,482-(int)(i*(((double)height/width))));
    }
    // Draw top-right corner
    for (int i=0; i<=490; i=i+10) {
        g.drawLine(i+255,167,745,167+(int)(i*(((double)height/width))));
    }
    /// Draw top-left corner
    for (int i=0; i<=490; i=i+10) {
        g.drawLine(745-i,167,255,167+(int)(i*(((double)height/width))));
    }

    }

/**
 * 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);
}

}

```