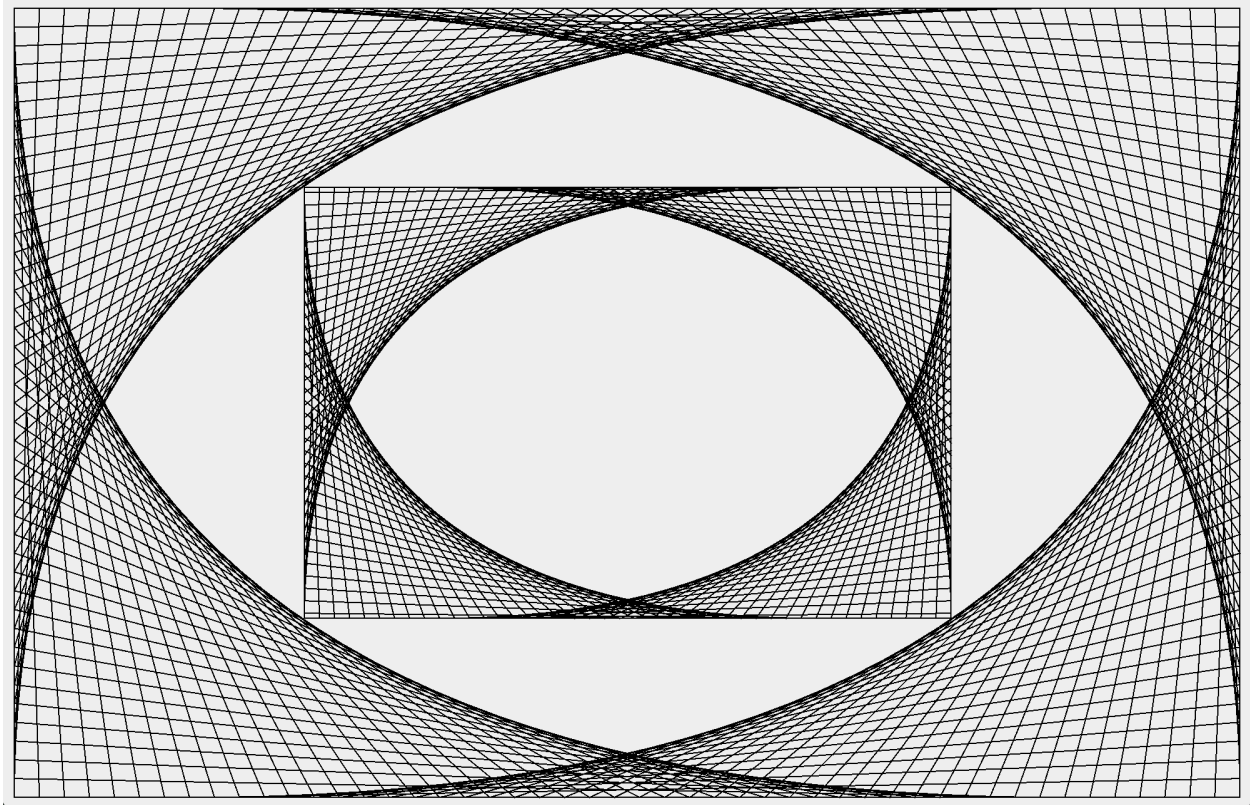




```
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
import javax.swing.JFrame;
public class LineArt extends JPanel {
// Unique version ID for this class to ensure saved objects can be loaded
safely
private static final long serialVersionUID = 1L;
// Initial width of height 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 x=990;
    int y=10;
    while (x>=20 && y<=640) {
        g.drawLine(x, 640, 990, y);
        x = x-20;
        y = y+15;
    }
    // Draw bottom-right corner
    x= 10;
    y=10;
    while (x<=990 && y<=640) {
        g.drawLine (x,640,10,y);
        x = x+20;
        y=y+15;
    }
    // Draw top-left corner
    int x2 = 10;
    int y2 = 640;
    while (x2<=990 && y2 >= 10) {
        g.drawLine(10, y2, x2, 10);
        x2 = x2+20;
        y2 = y2 -15;
    }
    // Draw top-right corner
    x2 = 990;
    y2 = 640;
    while (x2>=20 && y2>=10) {
        g.drawLine(990, y2, x2,10);
    }
}

```

```

x2 = x2-20;
y2 = y2 - 15;
}
// Tiny center copy (Extra credit)
g.drawRect(242, 153, 517, 344);
//bottom left corner (inner)
x=759;
y=153;
while(x>=242 && y<=497) {
g.drawLine(x, 497, 759, y);
System.out.println(x + " and " + y);
x=x-12;
y=y+10;
}
//bottom right corner (inner)
x = 242;
y = 153;
while (x <= 759 && y<=497) {
g.drawLine(x, 497, 242, y);
x = x+12;
y = y+ 10;
}
//top left corner (inner)
x= 242;
y= 497;
while(x<=759 && y>=153) {
g.drawLine(242, y, x, 153);
x=x+12;
y=y-10;
}
//top right corner
x=759;
y=497;
while(x>=242 && y>= 153) {
g.drawLine(759, y, x, 153);
x= x-12;
y= y-10;
}
}
}
/**
 * Overrides JPanel's paintComponent method to perform custom drawing.
 * 6
 * @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);
}
}
```