```java
import java.awt.Dimension;
import java.awt.Graphics;
import javax.swing.JPanel;
import javax.swing.JFrame;
public class LineArtLab06 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;
	private static int small_width = 560;
	private static int small_height = 360;

	// main method to launch the program as a standalone application - no need to
	// modify
	public static void main(String[] args) {
		LineArtLab06 panel = new LineArtLab06();
		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<=height; i+=10)
                {
                        g.drawLine(10+width, 10+i, (10+width)-i, 10+height);
                }

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

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

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


                // Draw a new rectangle
                int left_end = (1000-small_width)/2;
                int top = ((650-small_height)/2);

                g.drawRect(left_end, top, small_width, small_height);

                // Draw bottom-left corner
                for (int i = 0; i<=small_height; i+=10)
                {
                        g.drawLine(left_end+small_width, top+i, (left_end+small_width)-i,
top+small_height);
                }

                // Draw bottom-right corner
                for (int i = 0; i<=small_height; i+=10)
```

```
                {
                        g.drawLine(left_end, top+i, left_end+i, top+small_height);
                }

                // Draw top-right corner
                for (int i = 0; i<=small_height; i+=10)
                {
                        g.drawLine(left_end, (top+small_height)-i, left_end+i, top);
                }

                // Draw top-left corner
                for (int i = 0; i<=small_height; i+=10)
                {
                        g.drawLine(left_end+small_width, (top+small_height)-i,
(left_end+small_width)-i, top);
                }




        }
                /**
                * 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);
        }
}
```