```java
import java.awt.Color;
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) {
int x = 0;
int y = 0;
int point = 100;
// Draw the initial rectangle
g.setColor(Color.BLACK);
g.fillRect(10, 10, width, height);
// Draw bottom-left corner
g.setColor(Color.WHITE);
while(x <= 980 && y <= 630) {
     g.drawLine(990,10+y,990-x,640);
     x=x+((width/point)+1);
     y=y+((height/point)+1);
}
x=0;
y=0;
// Draw bottom-right corner
while(x <= 980 && y <= 630) {
     g.drawLine(10,10+y,10+x,640);
     x=x+((width/point)+1);
     y=y+((height/point)+1);
}
x=0;
y=0;
// Draw top-right corner
while(x <= 980 && y <= 630) {
     g.drawLine(990,640-y,990-x,10);
     x=x+((width/point)+1);
     y=y+((height/point)+1);
}
```

```
x=0;
y=0;
// Draw top-left corner
while(x<=980 && y<=630) {
     g.drawLine(10,640-y,10+x,10);
     x=x+((width/point)+1);
     y=y+((height/point)+1);
}
g.drawRect(244, 160, 511, 329);
int tall = 329;
int longg = 511;
x=0;
y=0;
while(x<=511 && y<=329) {
     g.drawLine(755,160+y,755-x,489);
     x=x+((longg/point)+1);
     y=y+((tall/point)+1);
}
x=0;
y=0;
// Draw bottom-right corner
while(x<=511 && y<=329) {
     g.drawLine(244,160+y,244+x,489);
     x=x+((longg/point)+1);
     y=y+((tall/point)+1);
}
x=0;
y=0;
// Draw top-right corner
while(x<=511 && y<=329) {
     g.drawLine(755,489-y,755-x,160);
     x=x+((longg/point)+1);
     y=y+((tall/point)+1);
}
x=0;
```

```
y=0;
// Draw top-left corner
while(x<=511 && y<=329) {
     g.drawLine(244,489-y,244+x,160);
     x=x+((longg/point)+1);
     y=y+((tall/point)+1);
}
g.drawRect(370,242,260,167);
tall = 167;
longg = 260;
x=0;
y=0;
while(x<=260 && y<=167) {
     g.drawLine(630,242+y,630-x,409);
     x=x+((longg/point)+1);
     y=y+((tall/point)+1);
}
x=0;
y=0;
// Draw bottom-right corner
while(x<=260 && y<=167) {
     g.drawLine(370,242+y,370+x,409);
     x=x+((longg/point)+1);
     y=y+((tall/point)+1);
}
x=0;
y=0;
// Draw top-right corner
while(x<=260 && y<=167) {
     g.drawLine(630,409-y,630-x,242);
     x=x+((longg/point)+1);
     y=y+((tall/point)+1);
}
x=0;
y=0;
```

```
// Draw top-left corner
while(x⩽260 && y⩽167) {
     g.drawLine(370,409-y,370+x,242);
     x=x+((longg/point)+1);
     y=y+((tall/point)+1);
}
x=0;
y=0;
int space = 0;
while (y⩽40 && x⩽40) {
     g.fillOval(450+x, 275+y, 100-space, 100-space);
     x=x+10;
     y=y+10;
     space= space+20;
     if(g.getColor()==Color.WHITE) {
          g.setColor(Color.BLACK);
     }
     else {
          g.setColor(Color.WHITE);
     }
}
}
/**
* 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);
}
}
```