

```
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Color;
import java.util.Arrays;
import java.util.Random;
import javax.swing.JPanel;
import javax.swing.JFrame;
```

```
public class Stars 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 window
    private static int width = 1000;
    private static int height = 650;
```

```
    // main method to launch the program as a standalone application - no need to modify
    public static void main(String[] args) {
        Stars panel = new Stars();
        panel.setPreferredSize(new Dimension(width, height)); // content size window
```

dimensions

```
        JFrame frame = new JFrame("Stars"); // Title of frame
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(panel);
        frame.pack();
        frame.setVisible(true);
```

```
    }
```

```
    /**
```

* Draws 10 stars (or more) of random sizes in random locations. The stars do not have to be fully contained within the panel.

```
    * @param g the Graphics object used for drawing shapes, text, and images
    */
```

```
    public void drawStars(Graphics g) {
        Random randy = new Random();
        double innerRadius;
        double outerRadius;
        for (int i = 0; i<10; i++) {
            int centerX = randy.nextInt(width);
            int centerY = randy.nextInt(height);
            int points = randy.nextInt(6)+4;
```

```

        innerRadius = (double) randy.nextInt(100)+11;
        outerRadius = (double) randy.nextInt(100)+21+innerRadius*1.25;
        int [] xpoints = new int [points*2];
        int [] ypoints = new int [points*2];
        for(int n = 0; n<(points*2); n++) {
            if (n%2 ==0) {
                ypoints[n] = centerY + (int) (innerRadius *
Math.sin(Math.toRadians(((360.0)/(points*2))*n)));
                xpoints[n] = centerX + (int) (innerRadius *
Math.cos(Math.toRadians(((360.0)/(points*2))*n)));
            }
            else {
                ypoints[n] = centerY + (int) (outerRadius *
Math.sin(Math.toRadians(((360.0)/(points*2))*n)));
                xpoints[n] = centerX + (int) (outerRadius *
Math.cos(Math.toRadians(((360.0)/(points*2))*n)));
            }
        }
        Color c = new
Color(randy.nextInt(256),randy.nextInt(256),randy.nextInt(256));
        System.out.println(Arrays.toString(xpoints));
        System.out.println(Arrays.toString(ypoints));
        System.out.println(innerRadius);
        System.out.println(outerRadius + "\n");
        g.setColor(c);
        g.fillPolygon(xpoints, ypoints, points*2);
    }
}

/**
 * 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
    drawStars(g);
}
}

```