

```

import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Color;
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) {

        //adjustable by users

```

```

for (int i = 0; i < 1000; i++) {

    int pointNum = (int)(Math.random()*95) + 5;

    int[] xPoints = new int[pointNum * 2];
    int[] yPoints = new int[pointNum * 2];

    int xPos = (int)(Math.random() * width);
    int yPos = (int)(Math.random() * height);

    int inCircle = (int)(Math.random() * 40) + 20;
    int outCircle = (int)(inCircle * 2);

    double vertexDistance = (2 * Math.PI)/(pointNum * 2);

    double radius = 0;
    for (int s = 0; s < (pointNum * 2); s++) {

        double angle = s * vertexDistance;

        if (s % 2 == 0) {
            radius = outCircle;
        }

        else {
            radius = inCircle;
        }

        xPoints[s] = (int)(xPos + radius * Math.cos(angle));
        yPoints[s] = (int)(yPos + radius * Math.sin(angle));

    }
    g.setColor(new Color((float)Math.random(),
(float)Math.random(), (float)Math.random()));

```

```

        g.fillPolygon(xPoints, yPoints, (pointNum * 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);
}
}

```

