

```

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

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 random = new Random();

        for(int j = 0; j<50; j++) {
            int small_r = random.nextInt(20, 50);
            int big_r = small_r*2;
            int r;
            int sides = random.nextInt(4,21);
            int points = sides*2;

```

```

        int[] xpoints = new int[points];
        int[] ypoints = new int[points];
        int x_offset = random.nextInt(0, width);
        int y_offset = random.nextInt(0, height);

        Color randomColor = new Color(random.nextInt(256),
random.nextInt(256), random.nextInt(256));
        g.setColor(randomColor);

        for (int i = 0; i<points; i++) {
            if(i%2==0){
                r=big_r;
            }

            else{
                r=small_r;
            }
            xpoints[i] = x_offset +
(int)(r*Math.cos(Math.PI/2+2*Math.PI*i/points));
            ypoints[i] = y_offset +
(int)(r*Math.sin(Math.PI/2+2*Math.PI*i/points));
        }

        g.fillPolygon(xpoints, ypoints, points);
    }

}

/**
 * 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);
}

}

```