

```
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 randy = new Random();
        for(int count = 0; count<randy.nextInt(500); count++) {
            int r = randy.nextInt(256);
            int gr = randy.nextInt(256);
            int b = randy.nextInt(256);
```

```

        Color color = new Color(r,gr,b);
        g.setColor(color);
        int points = 2*randy.nextInt(30)+10;
        int[] x = new int[points];
        int[] y = new int[points];
        int centerX = randy.nextInt(1000);
        int centerY = randy.nextInt(650);
        int r1 =randy.nextInt(100);
        int r2 = r1*2;
        int n= 1;
        for(int i=0;i<points;i+=2) {
            x[i]=(int) (centerX-r1*Math.sin(Math.toRadians(n*720/points)));
            y[i]=(int) (centerY+r1*Math.cos(Math.toRadians(n*720/points)));
            n++;
        }
        n=1;
        for (int i =1; i<points; i+=2) {
            x[i]=(int)
(centerX-r2*Math.sin(Math.toRadians((360/points)+(n*720/points))));
            y[i]=(int)
(centerY+r2*Math.cos(Math.toRadians((360/points)+n*720/points)));
            n++;
        }
        g.fillPolygon(x,y,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);
}
}

```