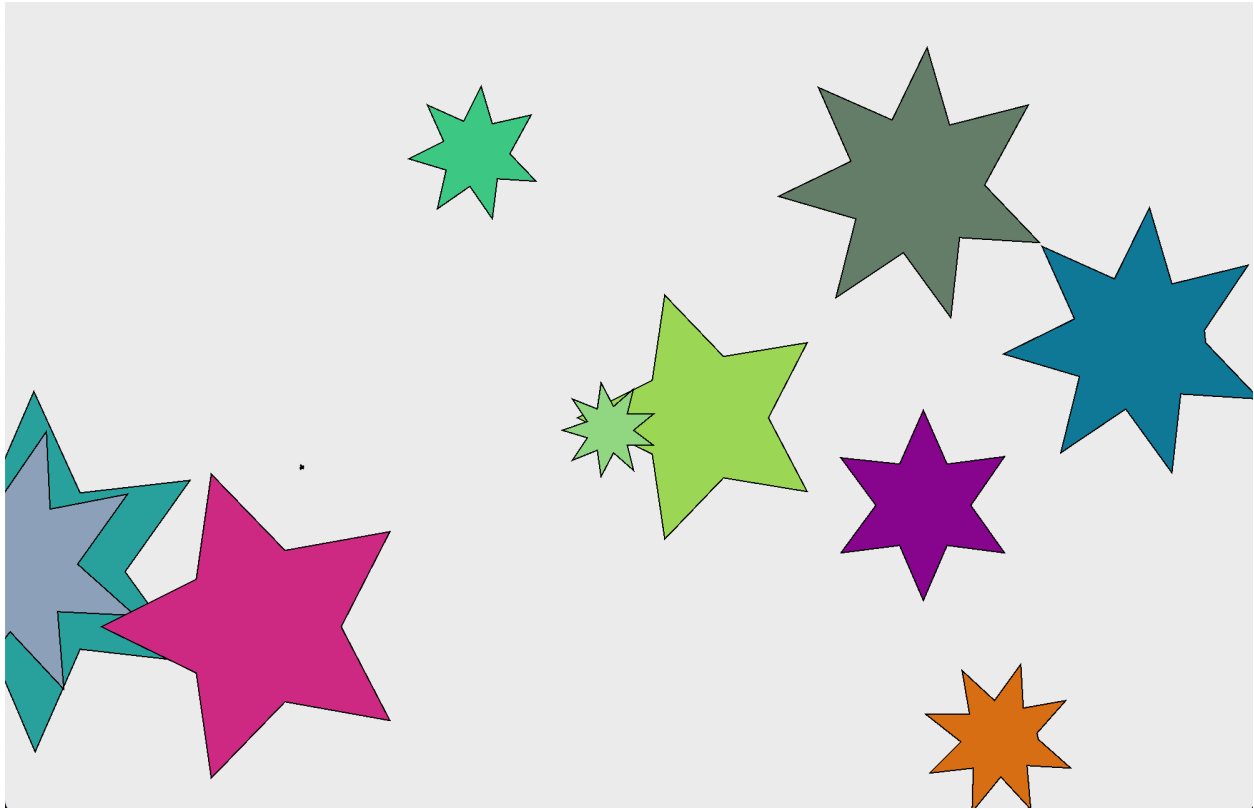




```
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Color;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.util.Random;

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 i = 0; i <= 10; i++) {
            int p = randy.nextInt(10,20);
            int r = randy.nextInt(75);
            int cx = randy.nextInt(1000);
            int cy = randy.nextInt(600);

            int [] x = new int [p];
            int [] y = new int [p];

            int n = 0;
            for (n = 0; n<p; n++) {
                if (n % 2 == 1) {
                    x[n] = (int)(cx+2*r*Math.cos(Math.toRadians(180/(p/2)*n)));
                    y[n] = (int)(cy+2*r*Math.sin(Math.toRadians(180/(p/2)*n)));
                }
                else {
                    x[n] = (int)(cx+(r*Math.cos(Math.toRadians(180/(p/2)*n))));
                    y[n] = (int)(cy+r*Math.sin(Math.toRadians(180/(p/2)*n)));
                }
            }

            Color c = new Color(randy.nextInt(255), randy.nextInt(255),
randy.nextInt(255));
            g.setColor(c);
            g.fillPolygon(x,y,p);
            Color black = new Color(0, 0, 0);

```

```
        g.setColor(black);
        g.drawPolygon(x,y,p);
    }
```

```
}
```

```
/**
 * 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);
}
```

```
}
```