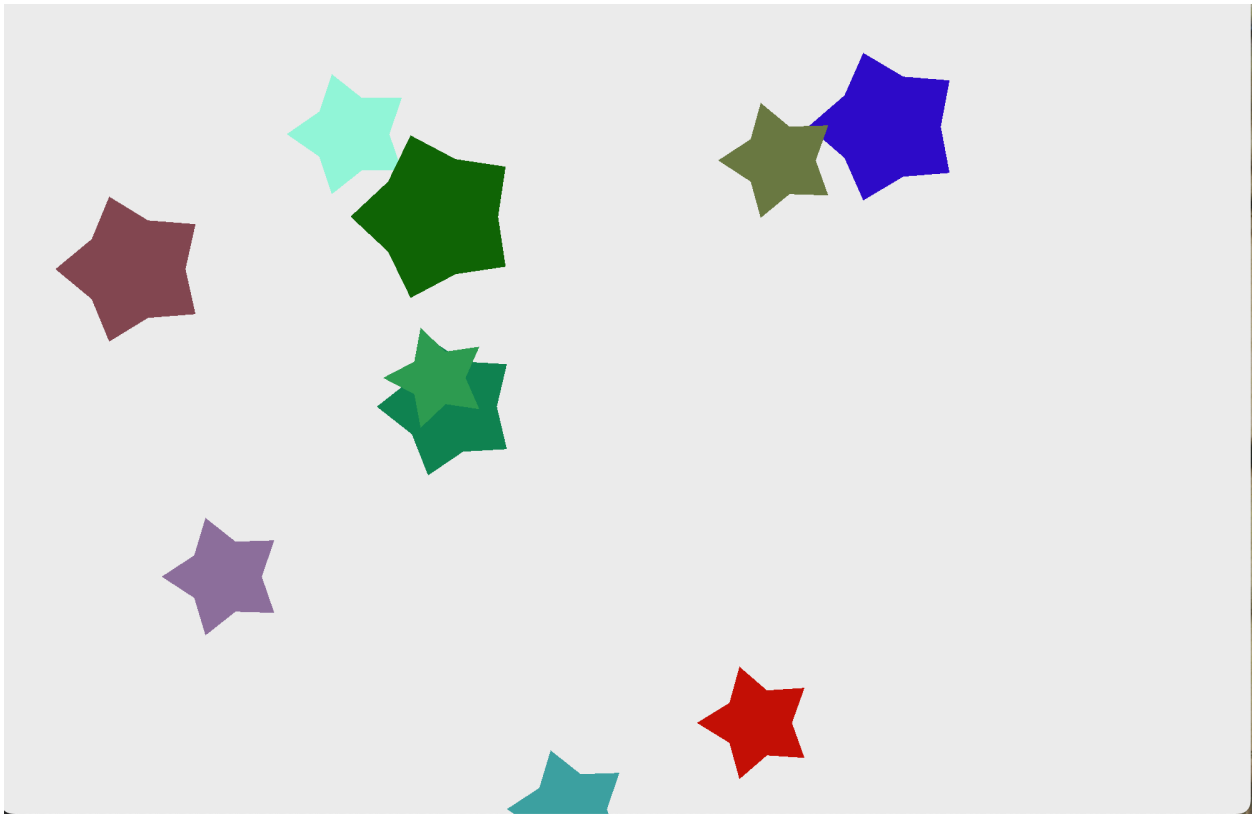

```
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Color;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.util.Random;
public class Stars extends JPanel {
// Unique version ID for this class to ensure saved objects can be loaded
safely
private static final long 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();
int NumStars = 10;
for (int s = 0; s < NumStars; s++) {
int radius = randy.nextInt(20, 50);
int cx = randy.nextInt (100,800);
int cy = randy.nextInt (100,800);
int [] yPoints = new int[10];
int[] xPoints = new int [10];
double angleStep = 2 * (Math.PI/ 10);
//inner points
for (int i=0; i< yPoints.length && i < xPoints.length; i+=2) {
double angle = i * angleStep;
xPoints[i] = cx + (int) (radius * Math.cos(angle));
yPoints[i] = cy + (int) (radius * Math.sin(angle));
}
//outer points
int radius2 = radius +20;
for (int i=1; i< yPoints.length && i < xPoints.length; i+=2) {
double angle = i * angleStep;
xPoints[i] = cx + (int)(radius2 * Math.cos(angle));
yPoints[i] = cy +(int) (radius2 * Math.sin(angle));
}
g.setColor(new Color (randy.nextInt(256), randy.nextInt(256),
randy.nextInt(256)));
g.fillPolygon(xPoints, yPoints, xPoints.length);
}
}
/**
* 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);
```

```
}
}
```