```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;
import java.util.Random;

public class DrawingStars extends Applet {
        private static final long serialVersionUID = 1L;

        public void paint(Graphics g) {
                Random rand = new Random();
                int[] xCords = new int[10];
                int[] yCords = new int[10];

                for (int t = 0; t < 10; t++) {
                        double radiusDiff = (rand.nextDouble() * 2.0)
+ 1.5;
                        int innerRadius = rand.nextInt(50) + 5;
                        double outerRadius = innerRadius *
radiusDiff;
                        int xOffset = rand.nextInt(980 - (int)
outerRadius);
                        int yOffset = rand.nextInt(680 - (int)
outerRadius);
                        int degreeOffset = rand.nextInt(360);
                        int red = rand.nextInt(256);
                        int green = rand.nextInt(256);
                        int blue = rand.nextInt(256);

                        if (xOffset - outerRadius < 0)
                                xOffset = (int) outerRadius;

                        if (yOffset - outerRadius < 0)
                                yOffset = (int) outerRadius;

                        for (int n = 0; n < 9; n = n + 2) {
                                xCords[n] = (int) (outerRadius *
Math.cos(Math.toRadians(18 + (72 * (n / 2 - 1)) + degreeOffset)))
                                                + xOffset;
                                yCords[n] = (int) (outerRadius *
Math.sin(Math.toRadians(18 + (72 * (n / 2 - 1)) + degreeOffset)))
                                                + yOffset;
                        }

                        for (int n = 1; n < 10; n = n + 2) {
                                xCords[n] = (int) (innerRadius *
Math.cos(Math.toRadians(54 + (72 * (n / 2 - 1)) + degreeOffset)))
                                                + xOffset;
                                yCords[n] = (int) (innerRadius *
Math.sin(Math.toRadians(54 + (72 * (n / 2 - 1)) + degreeOffset)))
                                                + yOffset;
```

```
                        }

                        Color pretty = new Color(red, green, blue,
200);
                        g.setColor(pretty);
                        g.fillPolygon(xCords, yCords, 10);

                }
        }
}
```