

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

public class Stars extends Applet {
    static final long serialVersionUID = 0L;

    static final int NUM_STAR_POINTS = 10;

    static final Color[] colors = {
        Color.BLACK,
        Color.BLUE,
        Color.CYAN,
        Color.DARK_GRAY,
        Color.GRAY,
        Color.GREEN,
        Color.LIGHT_GRAY,
        Color.MAGENTA,
        Color.ORANGE,
        Color.PINK,
        Color.RED,
        Color.YELLOW
    };

    public static Random rand = new Random();

    public static Color getColor() {
        return colors[rand.nextInt(colors.length)];
    }

    public static boolean drawStar(Graphics g, final int centerX, int
centerY, final double outerRadius,
        final int marginX1, final int marginY1, final int marginX2,
final int marginY2) {
        boolean starDrawn = false;

        double innerRadius =
outerRadius*Math.sin(Math.toRadians(18)/Math.sin(Math.toRadians(54)));

        int[] xCoordinates = new int[NUM_STAR_POINTS];
        int[] yCoordinates = new int[NUM_STAR_POINTS];

        // Note that (i-18)/36 will be 0, 2, 4, 6 8
        for (int i = 18; i < 360; i += 72) {
            xCoordinates[(i-18)/36] = centerX + (int) (outerRadius *
Math.cos(Math.toRadians(i)));
            yCoordinates[(i-18)/36] = centerY - (int) (outerRadius *
Math.sin(Math.toRadians(i)));
        }
    }
}
```

```

// Here (i-18)/36 will be 1, 3, 5, 7, 9
for (int i = 54; i < 360; i += 72) {
    xCoordinates[(i-18)/36] = centerX + (int) (innerRadius *
Math.cos(Math.toRadians(i)));
    yCoordinates[(i-18)/36] = centerY - (int) (innerRadius *
Math.sin(Math.toRadians(i)));
}

boolean validPoints = true;
for (int i=0; i < NUM_STAR_POINTS && validPoints == true; i++) {
    if (xCoordinates[i] < marginX1 || xCoordinates[i] > marginX2 ||
marginY2) {
        yCoordinates[i] < marginY1 || yCoordinates[i] >
        validPoints = false;
    }
}

if (validPoints) {
    g.setColor(getColor());
    g.fillPolygon(xCoordinates, yCoordinates, NUM_STAR_POINTS);
    starDrawn = true;
}

return starDrawn;
}

public void paint(Graphics g) {
    final int canvasWidth = 1000;
    final int canvasHeight = 700;
    final int margin = 10;
    final int marginX1 = margin;
    final int marginY1 = margin;
    final int marginX2 = marginX1 + (canvasWidth-(2*margin));
    final int marginY2 = marginY1 + (canvasHeight-(2*margin));

    setSize(canvasWidth, canvasHeight);

    Random rand = new Random();
    for (int i = 0; i < NUM_STAR_POINTS; ) {
        int centerX = rand.nextInt(canvasWidth) + margin;
        int centerY = rand.nextInt(canvasHeight) + margin;
        double outerRadius = (double) (rand.nextInt(200) + 25);

        if (drawStar(g, centerX, centerY, outerRadius, marginX1,
marginY1, marginX2, marginY2)==true) {
            i++;
        }
    }
}
}
}
}

```