**Part 1**

import java.awt.Color;

import java.awt.Graphics;

import java.util.Random;

public class Part1 {

public static int[] fillPolygonPoints(int nPoints) {

int[] array = new int[nPoints];

Random rand = new Random();

for(int i = 0; i < nPoints; i++) {

array[i] = rand.nextInt(400);

}

return array;

}

public static void fillPolygon(Graphics g, int nPoints) {

int[] xPoints = new int[nPoints], yPoints = new int[nPoints];

xPoints = fillPolygonPoints(nPoints);

yPoints = fillPolygonPoints(nPoints);

g.fillPolygon(xPoints, yPoints, nPoints);

}

public static void main(String[] args) {

int nPoints = 5;

DrawingPanel panel = new DrawingPanel(400, 400);

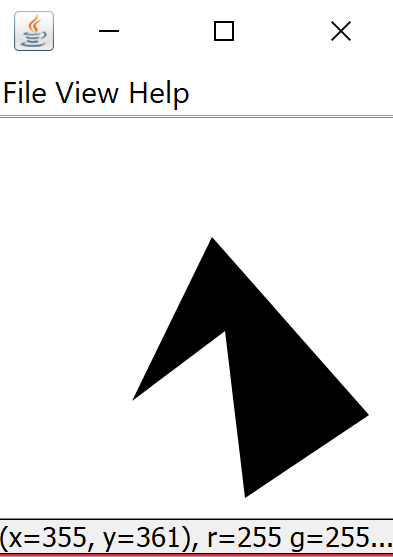
panel.setBackground(Color.WHITE);

Graphics g = panel.getGraphics();

fillPolygon(g, nPoints);

}

}



**Part 2**

public static void fillRegularPolygon(Graphics g, int ctrX, int ctrY, int radius, int nPoints) {

int[] xPoints = new int[nPoints], yPoints = new int[nPoints];

double angle = 360 / nPoints;

for(int i = 0; i < nPoints; i++) {

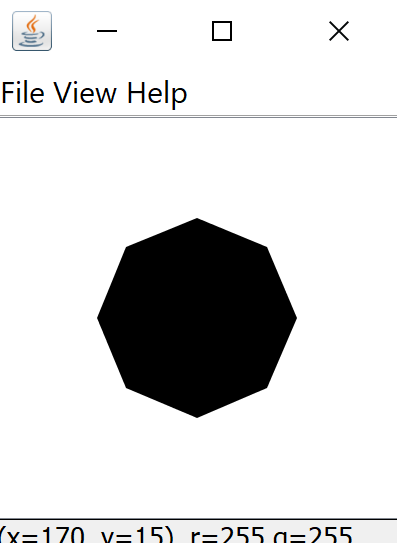
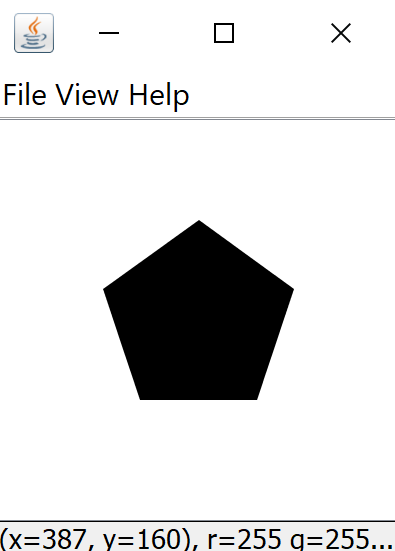
xPoints[i] = (int)(ctrX + radius \* Math.cos(Math.toRadians((angle \* i - 90))));

yPoints[i] = (int)(ctrX + radius \* Math.sin(Math.toRadians((angle \* i - 90))));

}

g.fillPolygon(xPoints, yPoints, nPoints);

}



**Part 3**

public static void fillStar(Graphics g, int ctrX, int ctrY, int radius, int nPoints, double spikiness) {

nPoints += nPoints;

int[] xPoints = new int[nPoints], yPoints = new int[nPoints ];

double innerRadius = radius \* (1.0 - spikiness);

double angle = 360 / nPoints;

for(int i = 0; i < nPoints; i += 2) {

xPoints[i] = (int)(ctrX + radius \* Math.cos(Math.toRadians((angle \* i - 90))));

yPoints[i] = (int)(ctrY + radius \* Math.sin(Math.toRadians((angle \* i - 90))));

}

for (int i = 1; i < nPoints; i+=2) {

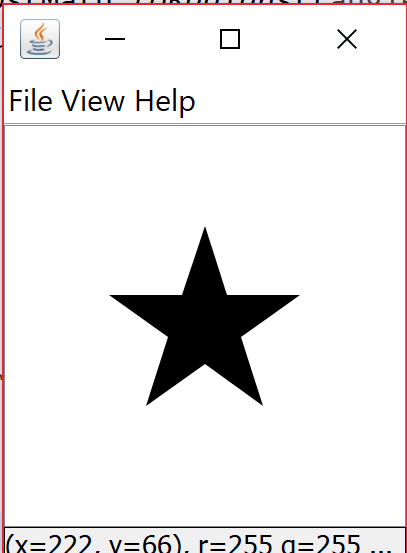
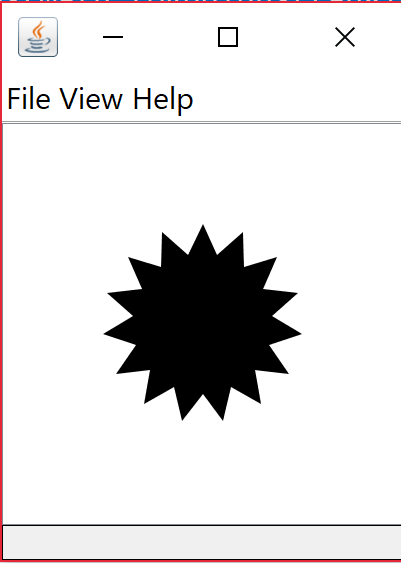
xPoints[i] = (int)(ctrX + innerRadius \* Math.cos(Math.toRadians((angle \* i -90))));

yPoints[i] = (int)(ctrY + innerRadius \* Math.sin(Math.toRadians((angle \* i -90))));

}

g.fillPolygon(xPoints, yPoints, nPoints);

}



**Part 4**

public static void main(String[] args) {

int nPoints = 29;

DrawingPanel panel = new DrawingPanel(2000, 2000);

panel.setBackground(Color.WHITE);

Graphics g = panel.getGraphics();

Random rand = new Random();

for(int i = 1; i <= 5; i++) {

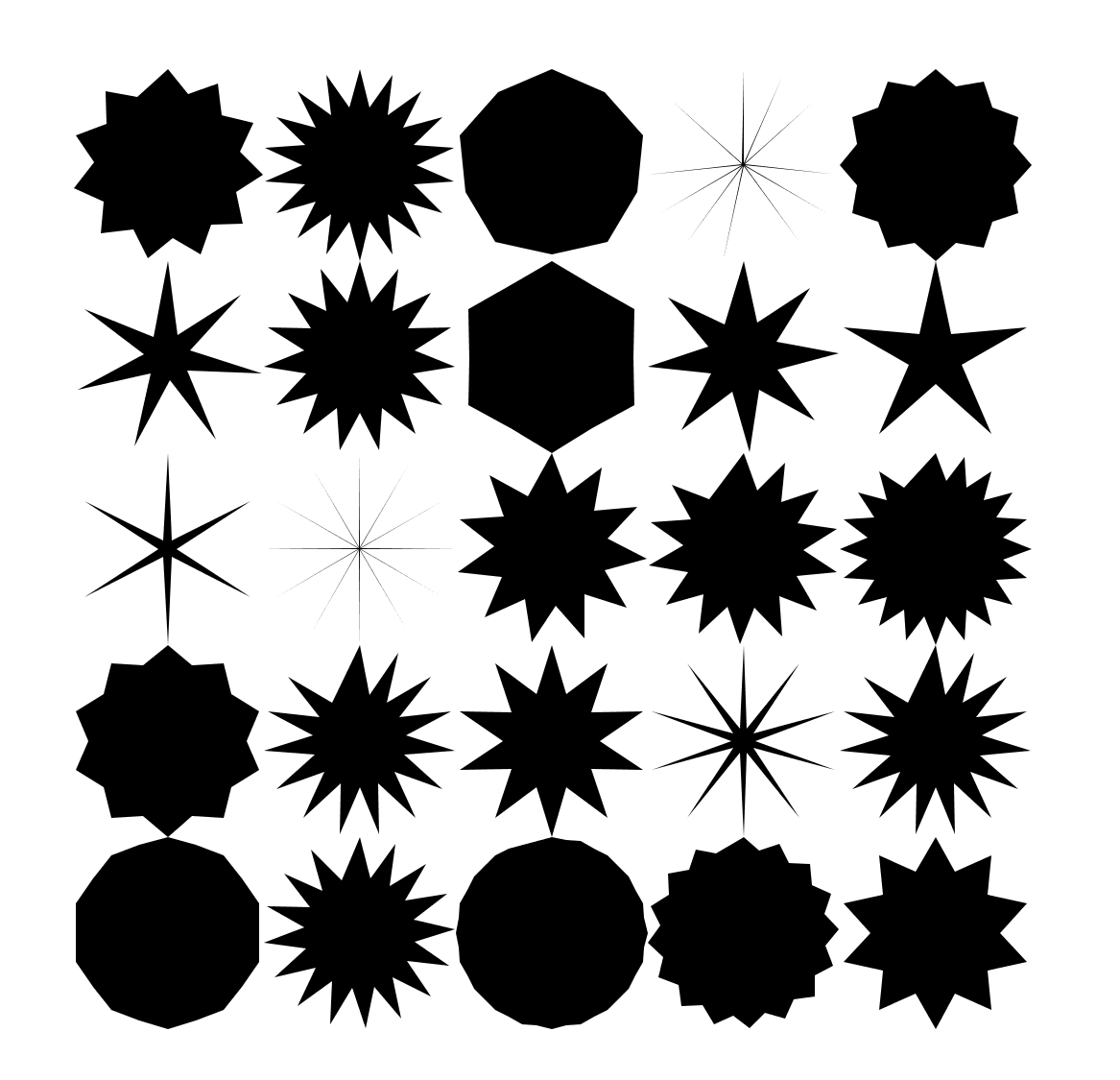
for(int j = 1; j <= 5; j++) {

fillStar(g, 200 \* j, 200 \* i, 100, rand.nextInt(15) + 5, rand.nextDouble());

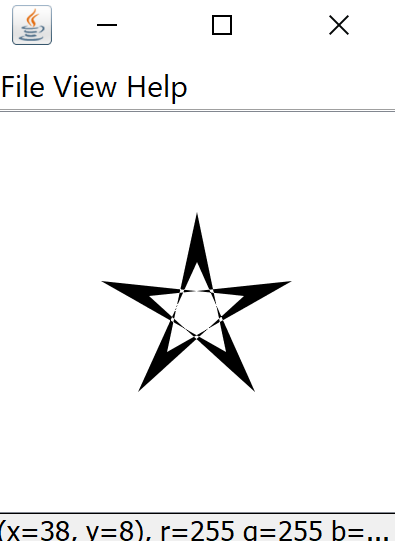
}

}

}

****

**Extra Credit**



The reason the stars value changes the way it does is because the inner radius value is now a negative value. Looking at the debugger the second x point gets set to the left of the starting point, rather than the right. In the assignment of the x point the innerRadius \* the cos of the angle becomes negative, the end result draws a line from the right side of the start to the left side, because the inner points where set in reverse order.