# HW2. Draw Curves from Straight Lines

Carefully read this *entire document* before beginning your work.

## Objectives

In this program, you’ll get in the introduction to using objects; we won’t create our own just yet (that comes later). You’ll also learn about and use typical graphics coordinate systems. In addition, you’ll create and use your own functions.

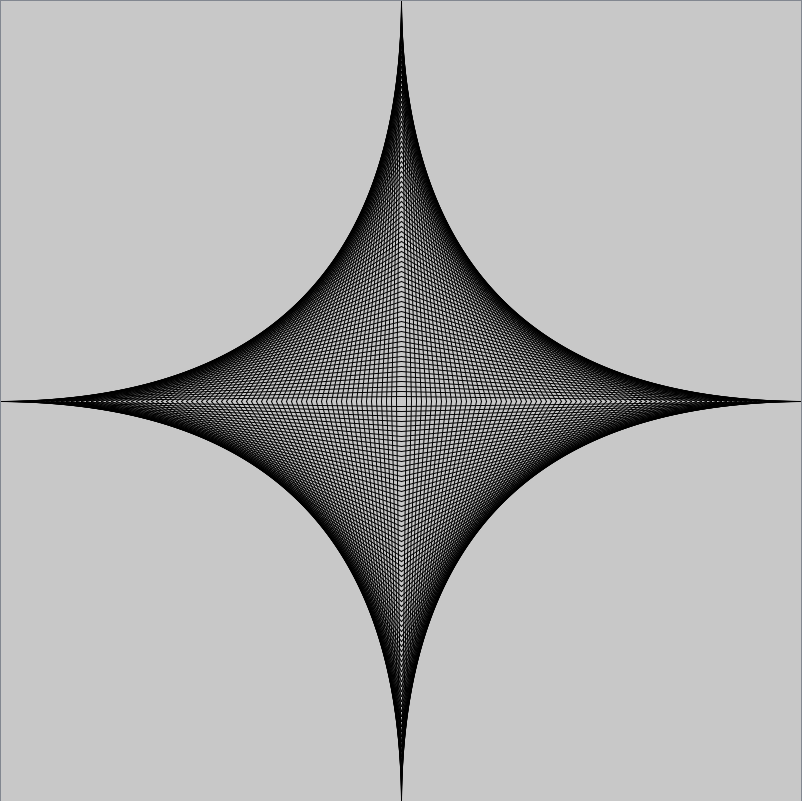
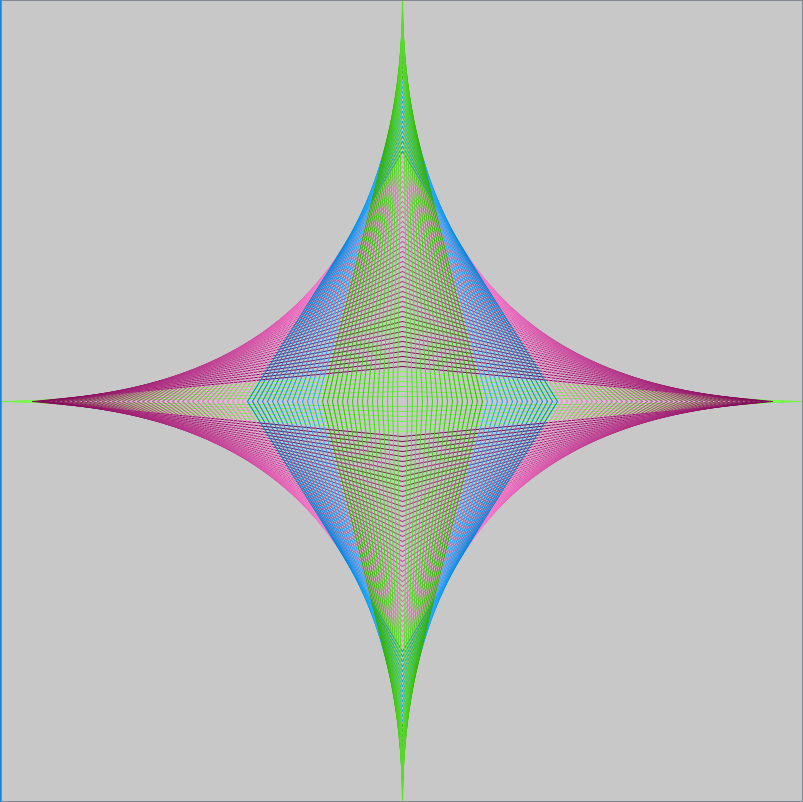
## Input

This program gathers no input from the user.

## Output

3.1 Part 1

* Generate this graphic output shown at right, on a 500x500 DrawingPanel.
* Alter the colors (drawing and background) to suit your own taste.
* For descriptions of the technique and samples of output, see these sites: [site1](http://mathematicsrealm.blogspot.com/2012/07/curves-formed-from-straight-lines.html), [site2](https://mathcraft.wonderhowto.com/how-to/create-parabolic-curves-using-straight-lines-0131301/).

3.2 Part 2

* Create an additional drawing panel at least 400 x 400. On it, draw filled-in shapes of at least two types (rectangles, circles, etc.) in at least two colors.
* Each shape type should be drawn by its own method (that you write) with parameters indicating where it is to be drawn and how big it is to be. Consider including the color as a parameter as well; this would make the function more flexible.
* Draw at least ten shapes in an attractive pattern; do something fun and interesting of your own design.

## Instructions

* Use the provided DrawingPanel.java class. Put this in the same folder as your solution.
* Use procedural decomposition to break down the program into logical pieces.
* Part 1: set constants for the drawing panel size and the line increment. Assume your figures may drawing on different sizes of the canvas.
* Part 1: try various line increments until you find one that is visually pleasing and looks curved.
* Use a Random object or random() method for creating different colors.

## Code Implementation

Create a class called **HW2Drawing**; use this single class to do all your work. You should have only one main() method that directs both parts of the project. Follow the provided ***Course Style Guide***.

5.1 What You Should Use

* Constants and variables
* Definite loops (For Part 1, a single loop should be used to draw the entire figure)
* Drawing panel and graphics objects
* A ***single*** main program that draws both drawings (Part 1 and Part 2)

5.2 What You Shouldn’t Use

* Selection control structures, unless you are doing something way “above and beyond” the specification.

## Submitting Your Work

Submit your **.jar** file; there should be ***DrawingPanel.java*** and only one your class.

## Hints

* Look over the supplied sample code that uses the DrawingPanel.
* Having a link to the online Java API reference will come in handy now and in the future.
* Think about how many loops you need; it may be way fewer than you think, if you’re clever. Make your code concise. Don’t duplicate code if it is avoidable.

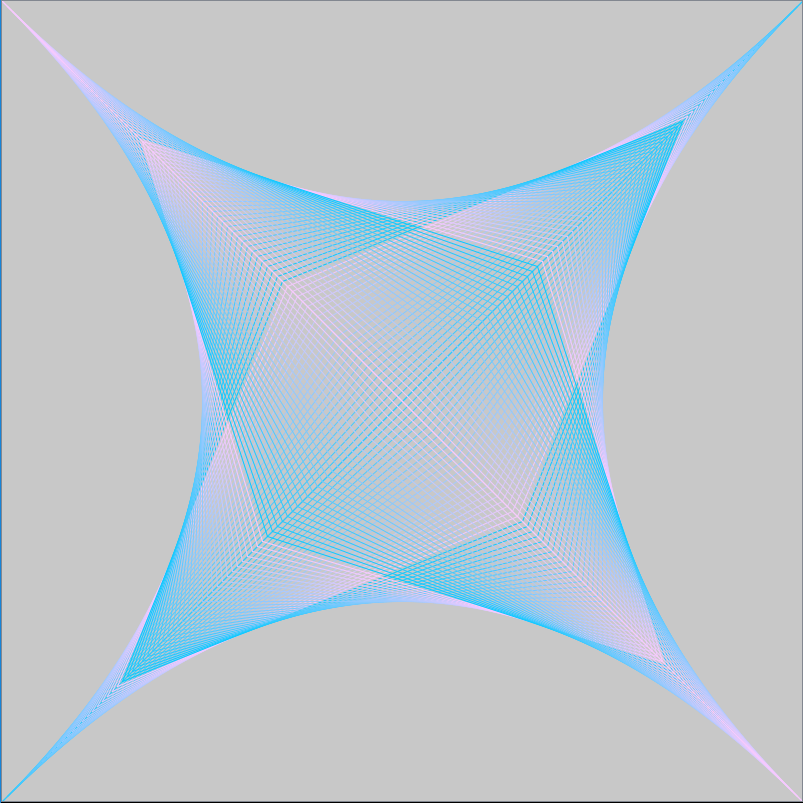
**Useful references:**

<https://www.muminthemadhouse.com/maths-and-art-collide-parabolics-curves/>

<https://www.mrsmilewski.com/parabolic-curve.html>

## Extra Credit (5 points)

Change the orientation of figure as shown on the following picture.



## Grading

