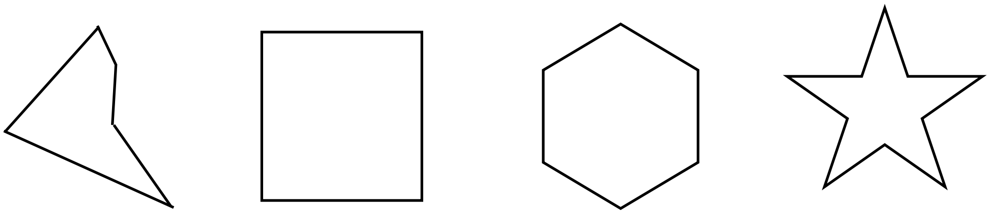
Milestone 1: Draw a `Polygon`.

The spaceship and asteroids are all going to be drawn on screen as polygons. Recall that a polygon is a chain of line segments that close to form at loop. Some examples of polygons are below.



For the first milestone, you will write the method to draw instances of class `Polygon` on the screen, and draw three such instances to test it out.

Detailed instructions:

Look at the class `Polygon`. An instance of this class represents a polygon, and its instance variables store the following information:

- the shape of the `Polygon`, represented as an array of `Point` instances corresponding to the vertices (corners)
- the offset between the origin and the center of the `Polygon` instance, represented as a `Point`
- the rotation in degrees of `Polygon` instance from its starting orientation (as given by the shape array)

What are the above instance variable names?

You'll notice that coordinates are represented by instances of the `Point` class. Look at the `Point` class to see what its instance variables are, and what methods it has.

To draw a `Polygon` on the screen, create a non-static method `paint(Graphics brush)` in `Polygon` that returns nothing. We will call this method from the `paint` method in `Asteroids`.

We will use the methods in the class `Graphics` (https://docs.oracle.com/javase/8/docs/api/java/awt/Graphics.html) to draw the `Polygon` instance. All of the methods in `Graphics` are non-static, so we call them by `brush.nameOfMethod(..)`, where `brush` is the parameter of our new `paint` method.

Before we draw the polygon, we'll make sure the `paint` method is set up correctly by using the `drawLine` method in `Graphics` to draw a single line between the coordinates (10,20) and

(30,40). To do this, type `brush.drawLine(10,20,30,40);` in the `paint` method in `Polygon`.

To test this code, you need to create an instance of `Polygon` in `Asteroids` (it does not matter what values are passed into the construct, as long as they are the right type). Then use this instance to call the method `paint` (in `Polygon`) from inside the method `paint` in `Asteroids`. The same `Graphics` instance (`brush`) that is a parameter in `paint` in `Asteroids` should be passed into `paint` in `Polygon`.

Once you have a line displaying on your screen, change `paint` (in `Polygon`) so that it draws the actual `Polygon` instance. The `getPoints()` method in `Polygon` will return an array of `Points` representing the vertices of the `Polygon` at its current position (i.e. with the `offset` and `rotation` applied). Once you have these coordinates, you can use methods in `Graphics` (https://docs.oracle.com/javase/8/docs/api/java/awt/Graphics.html) to draw the corresponding polygon. Note that at one of the methods in `Graphics` takes in a `Polygon` object, but it is one that is pre-defined in Java, not the one you are currently writing code for. Also, note that none of these methods take our `Point` objects, so you will have to extract the relevant information from them to pass into the `Graphics` method.

Test your code by drawing one or more polygons on the screen.