

Hyeon-Seung Yu's CS 151 Project 1: What is a Shape?

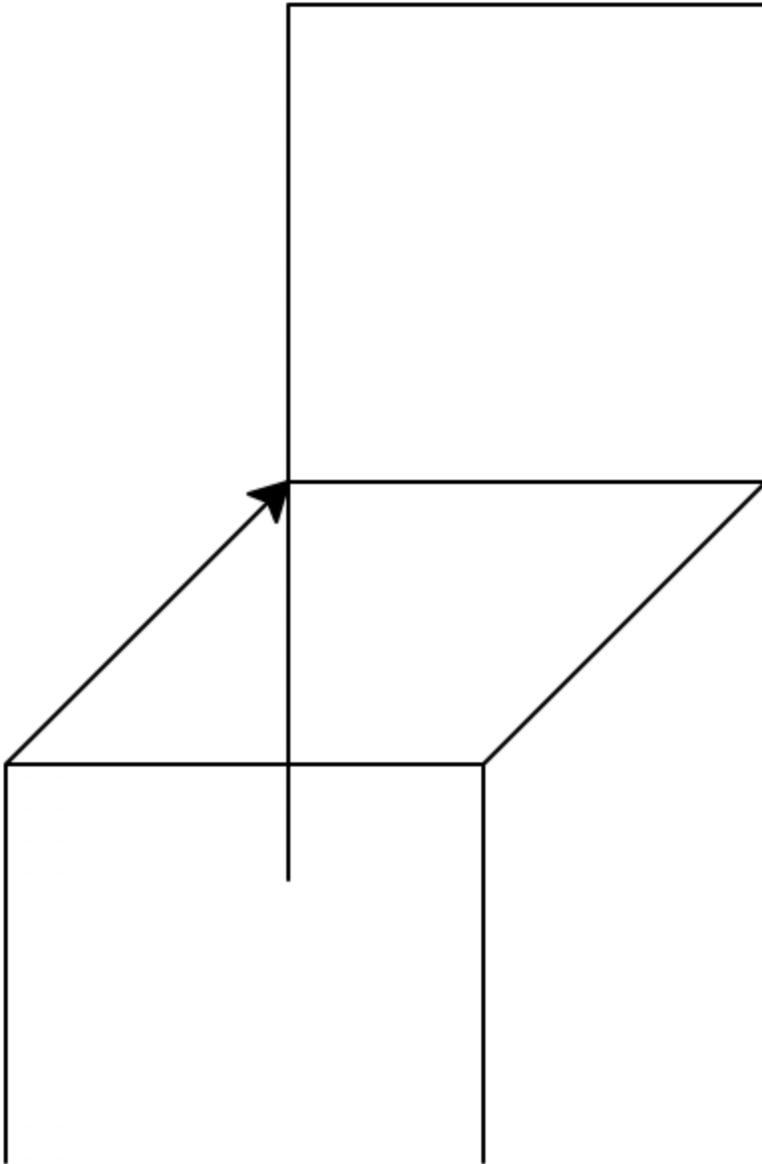
The purpose of this project was to learn how to use the 'turtle' feature of python programming language. As the name of the project suggests, the python turtle allows me to draw different shapes and create further complex shapes using different features. Descriptions and images below will illustrate how my shapes became more sophisticated as I moved on to different steps and extensions in the project.

Required Codes:

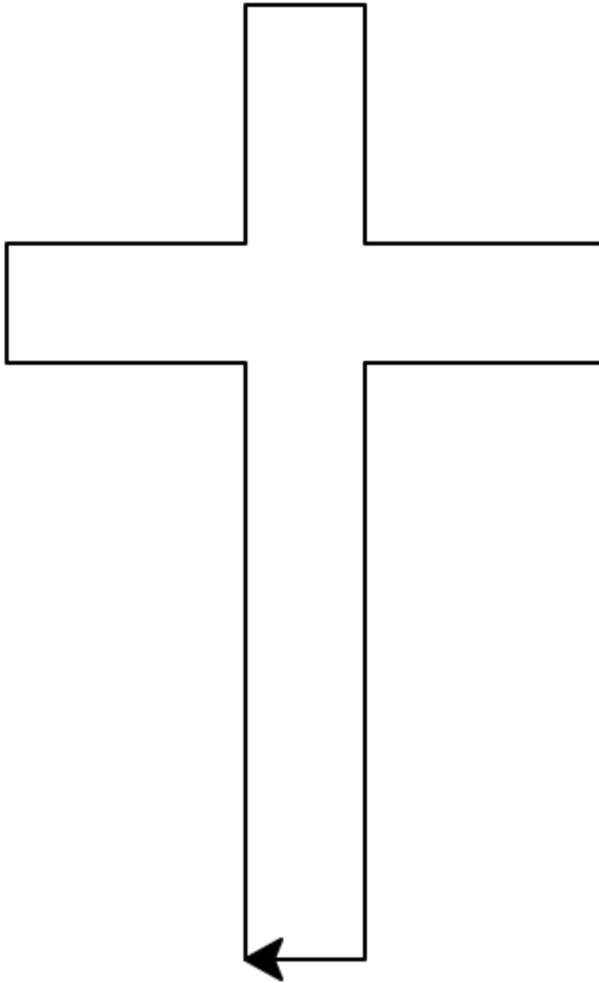
- shapesA.py draws a simple chair by moving the "turtle" to left and right at different distances and different angles. The snippet below shows some comands.

```
left(90)
forward(120)
right(90)
forward(120)
right(90)
```

- The image below is how shapeA looks like.



- shapeB.py draws a cross by moving the turtle left and right at different distances and angles. Basically both shape A and B are quite simple. Below is the image of shape B.



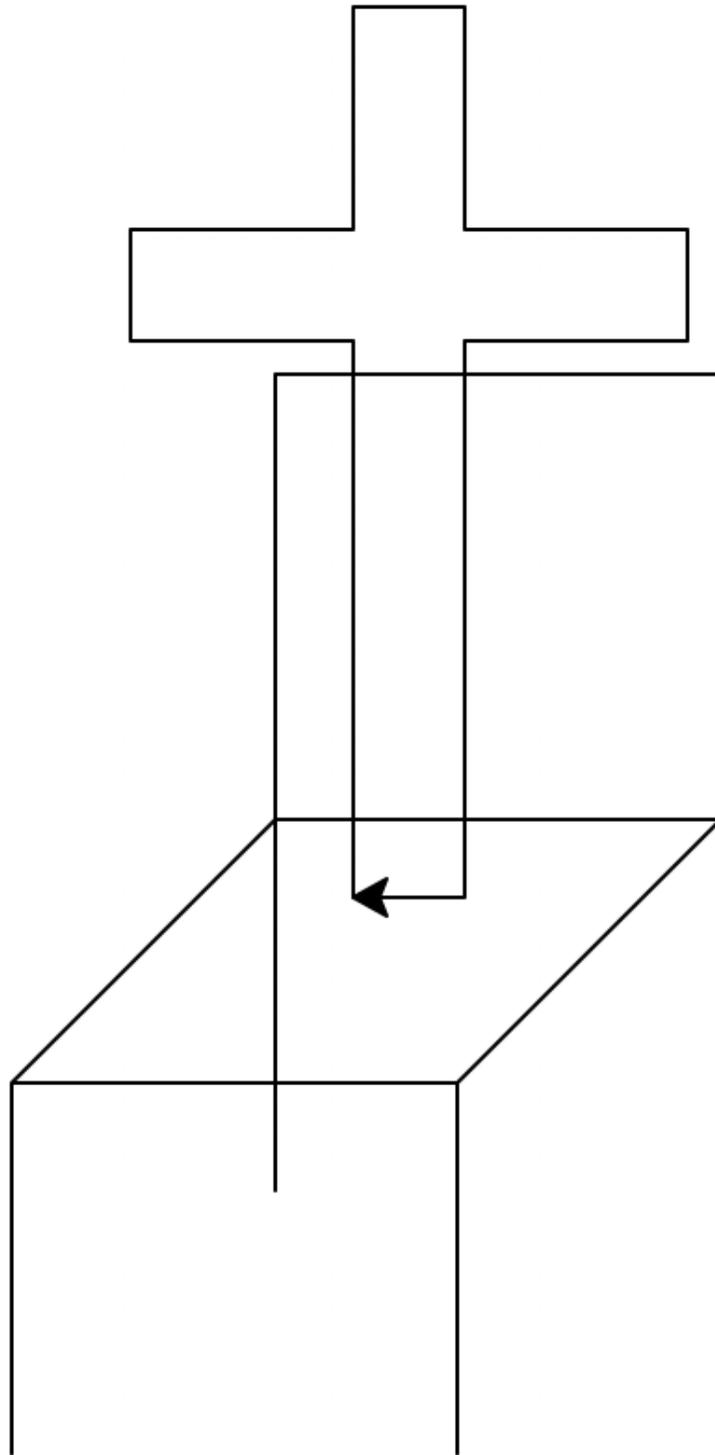
- In shapes.py, my codes and shapes start to become more complicated. First of all, I created multiple functions that define shapeA and shapeB so that I would not have to rewrite all the codes of shape A&B every time I want to draw them. The define function looks like below.

```
def shapeA():  
    left(90)  
    forward(120)  
    right(90)  
    forward(120)
```

- In shapes.py I also created a function that draws both shape A&B but there is a small movement of turtle between drawing those two shapes. I intentionally put some commands between shape A&B so that they would look awkward when drawn together. The first image is the commands that arranges the position of the turtle before drawing shape B. ShapeC looks like below.

```
up()  
right(90)  
forward(30)  
left(45)  
down()
```

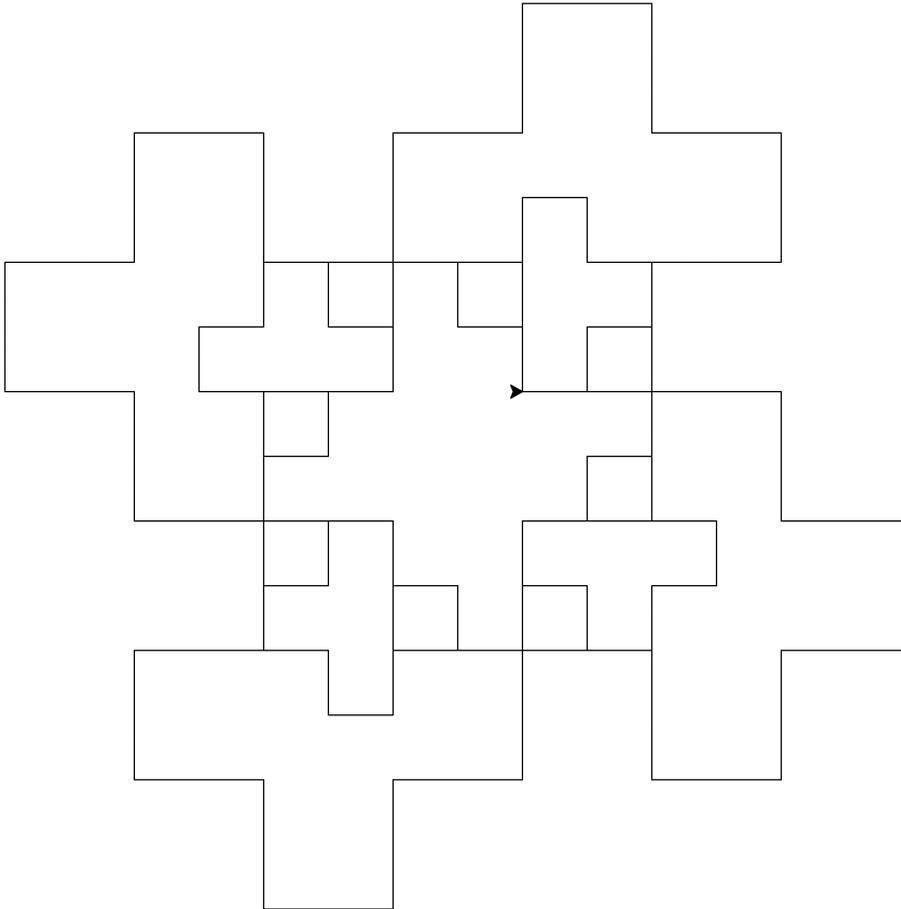
.



- Another function `shapeD` draws `shapeB` but with a parameter. what this means is that I can now draw `shapeB` with different length of sides whenever I change the number inside the function parenthesis. If you can notice, I wrote 'distance' next to the function.

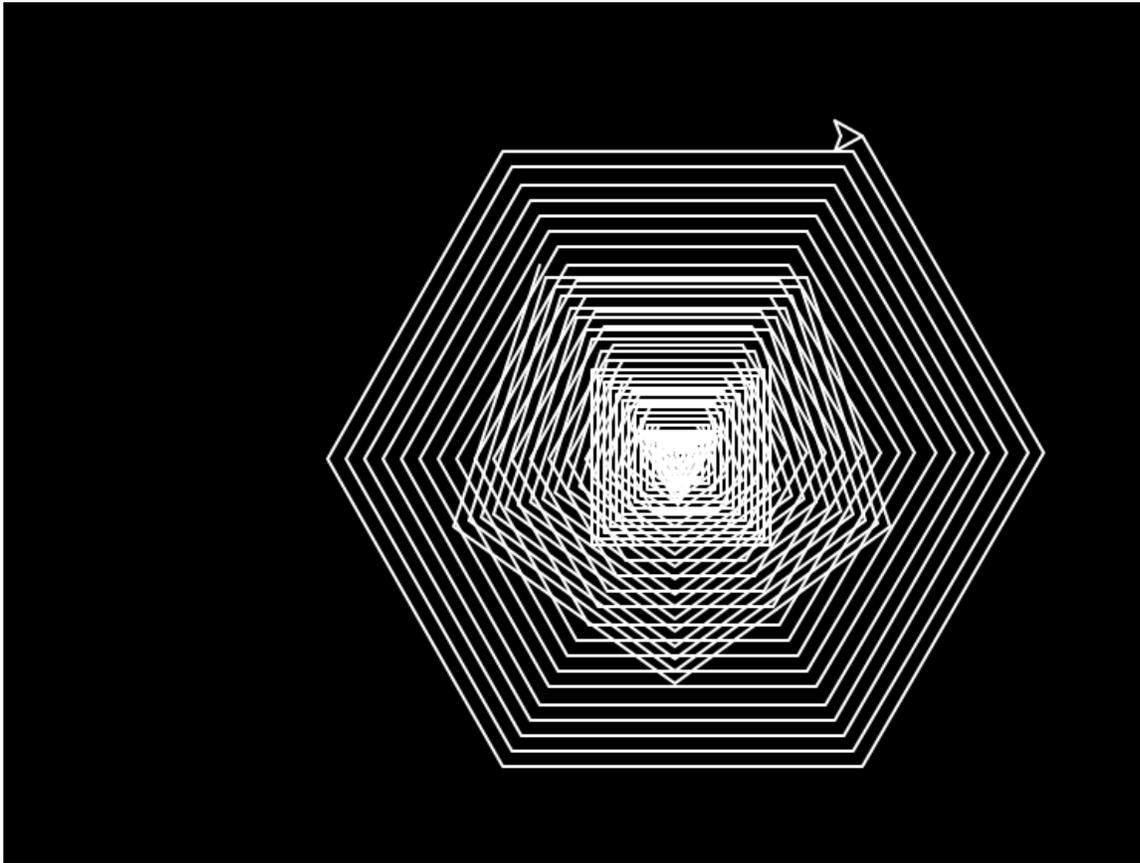
```
def shapeB(distance):  
    left(90)
```

- Now another shapeE draws shapeD four times at different locations. Similar with shapeD, shapeE will draw the shape of cross with the different length of sides due to the function parameter. It looks like the below image.

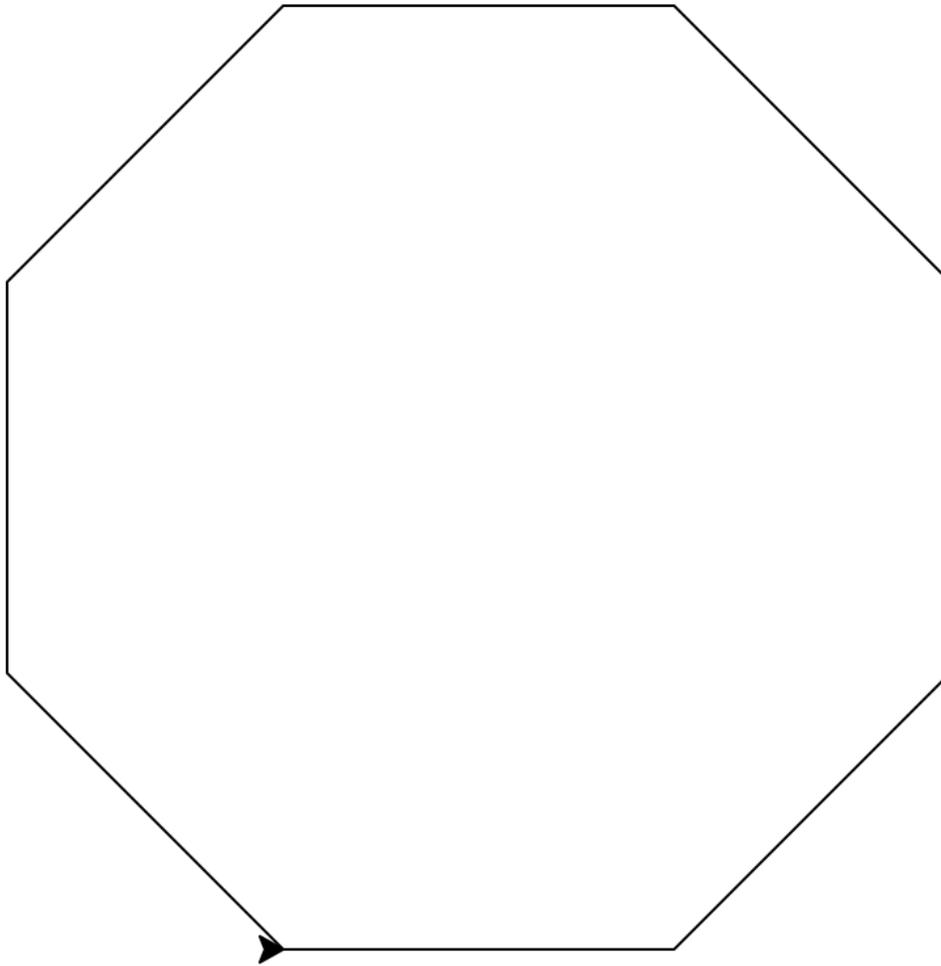


Extensions:

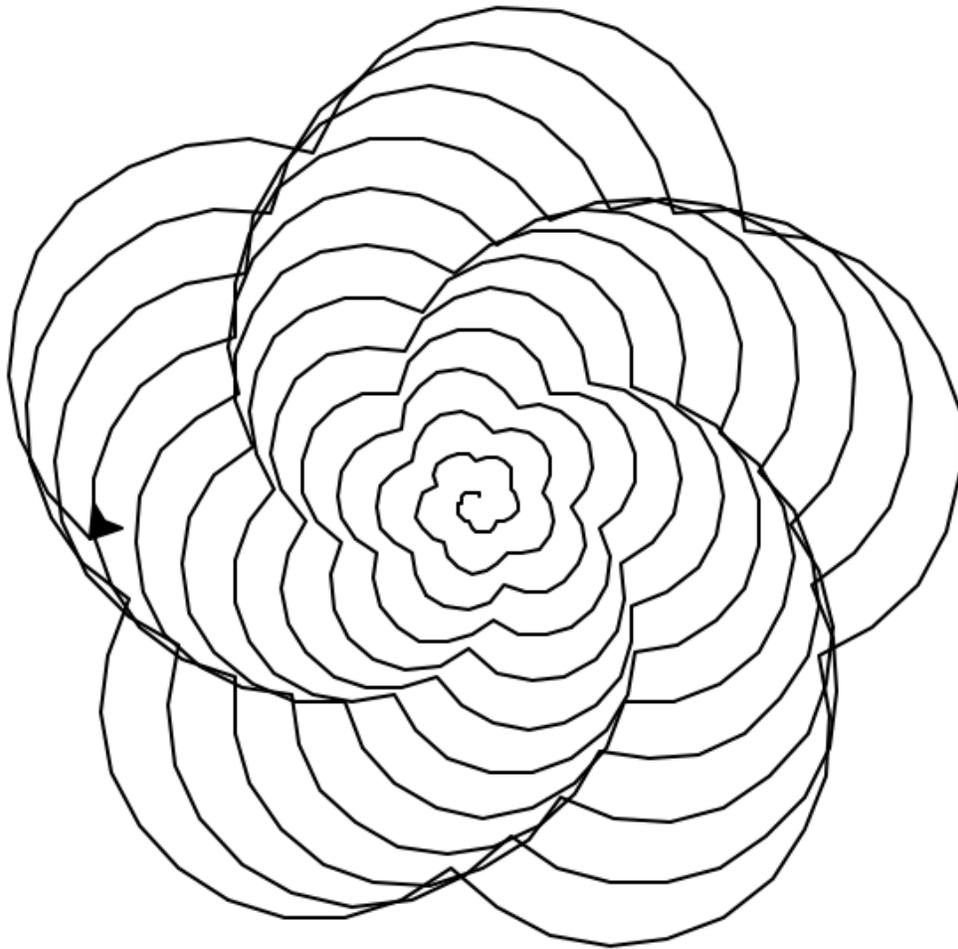
- extension1.py: I completed writing this code after finishing all other extensions; therefore, I am quite familiar using loop and range function. I decided to use those functions for this extension and wanted to draw polygons that would become bigger and increase the number of sides gradually. So I made the straight lines to bend at different angles to form triangle, square, pentagon, and hexagon. As the image below would show, the area of the shapes will increase as the shape turns into the next one because of the loop and range function. There are two more things to note about this shape: 1) Before the turtle draws the next polygon, it comes back to the original position because of the command ht(). I used this command so that the polygons would overlap with each other. 2) I set the background and pen's color as black and white so that the distinction between the polygons can become clearer. The shape looks like below.



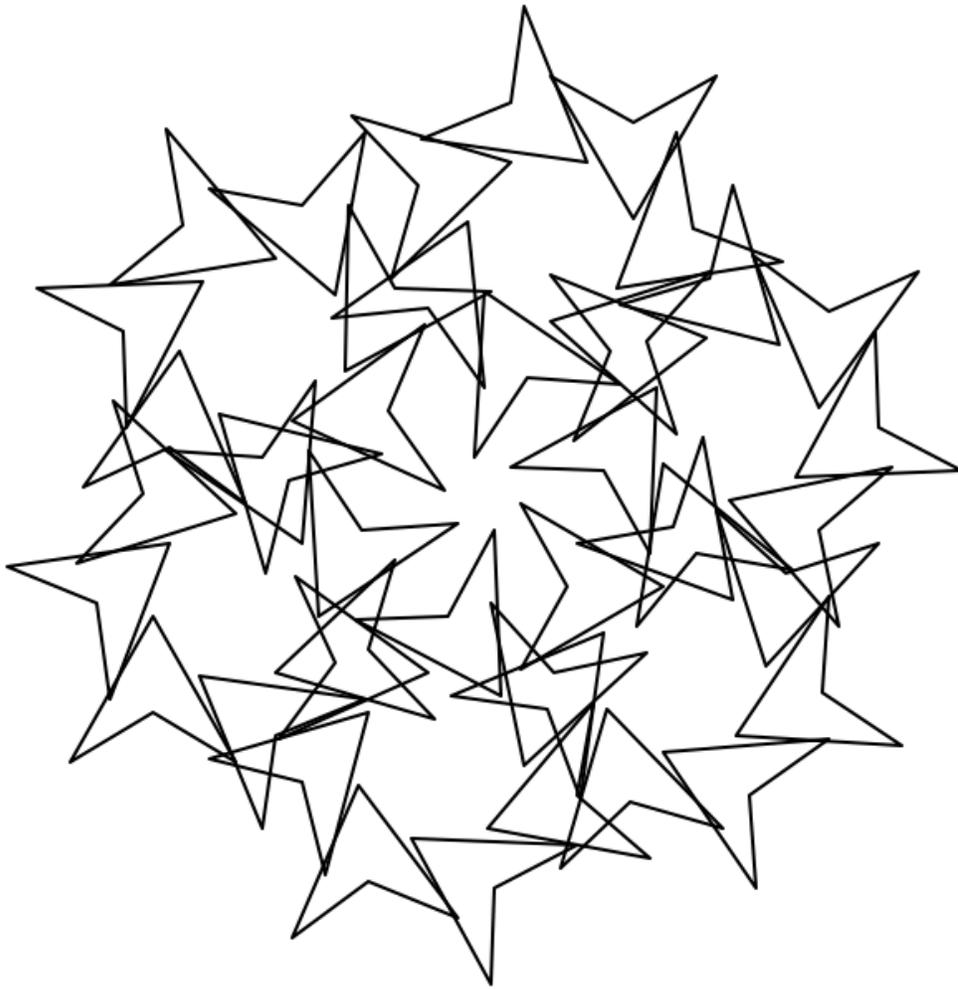
- extension2.py: This code has a function npolygon, which obviously draws a polygon with certain number of sides. This function uses parameter feature to draw n-polygons by setting two inputs: the length of sides and the number of sides. I guess one needs to know basic geometry to use this function because one should be aware that $360/\#$ of sides will determine the angle of the polygon. Anyhow, this is how extension2.py looks like; it looks like a octagon because I put 8 as the # of sides.



- extension3.py: Like extension1 and 2, this code uses loop and range function in order to repeat a certain command to create a unique graphic. I decided to use the command circle() so that I could draw arcs instead of plain straight lines. Although I did not have the idea of the end-result in my head when typing the codes, the shape is quite nice.(it looks like a flower?) One thing to note is that I did put 'i' in parameter on purpose so that the length of arc would get longer each time.



- extension4.py: Extension4 is quite similar with shapeE in a sense that they both draw a same shape multiple times and at different locations. Again, the result is quite different from the original shape. One thing to note is that the original shape – pointer-looking shape – switches its angle each time in order to form a circle at the end. This was done by loop and range function and putting specific angle.



Conclusions: I learned several lessons from this project since I am a beginner in computer programming. First of all, I realized codes can get very long very quickly. Then, I noticed there are different functions that can be used to reduce 'duplications' as Professor Skrien mentioned in class. I personally think the purpose of this project was to introduce the basics of python turtle and the concept of a function and its benefit in coding. While working on this project, I received help from Professor Skrien and Eaton, TAs in the weekends, most of all the interactive online textbook. I believe it was chapter 4 that helped me to figure out how to use loop and range functions. The example codes on the textbook also helped me to come up with my own for the project.