

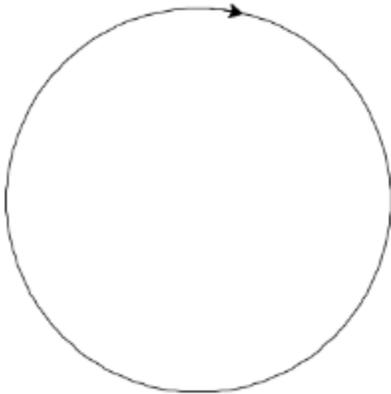
# cs151s20project1

## Karl Hick - Project 1

### Abstract:

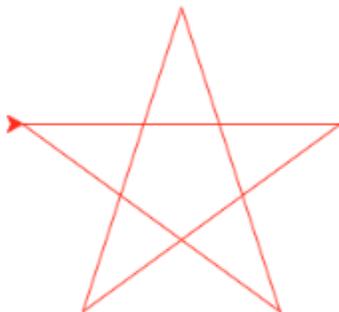
The key purpose of the assignment was to familiarize the student with basic commands in python and in the terminal, such as with giving simple commands and creating simple functions while making use of parameters. While using the turtle program, the student could define shapes and strings of commands in order to make use of them multiple times. The project began by asking for me to develop several simple shapes, combine those shapes, and then manipulate them in various ways.

### Task 1: shapeOne.py



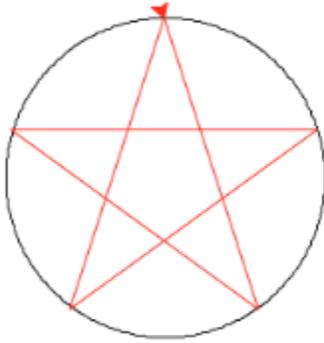
In order to describe the circle, I wrote 'forward (5) right (2),' copied and pasted it, and copied and pasted that in order to exponentially write out the requisite number (180) of line segments needed to make a full circle.

### Task 2: shapeTwo.py



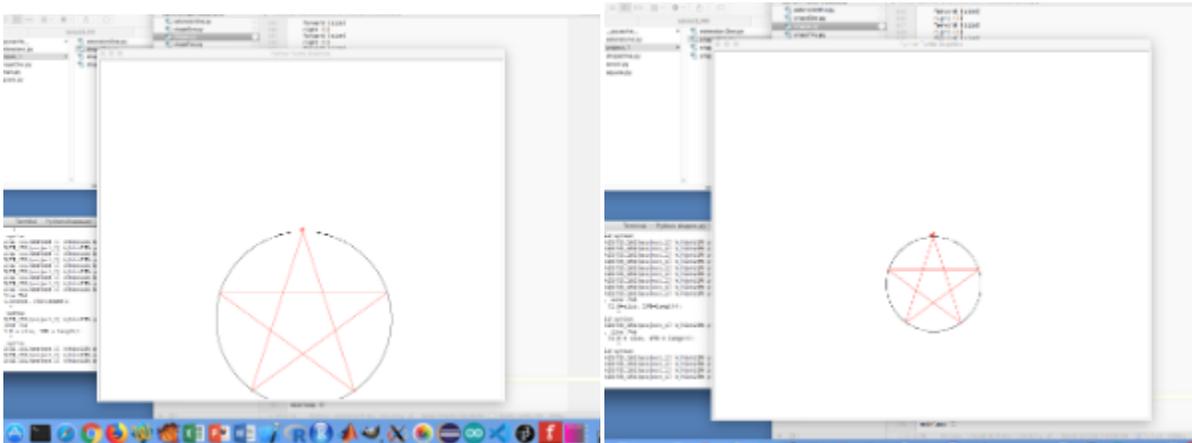
I wanted to make a suitable additional shape, though nothing came to mind outside of triangles and multi-sided shapes, which plagued my consciousness after I glanced at the list of extensions, so I combined the two into a simple star. I knew stars had an inner angle of  $36^\circ$ , which meant that I had to put in five turns of 'right (154)' to make tat turn from the outside angle. The star is red because I came up with a masterful combination midway through that I am planning for my third shape.

### Task 3: shapeThree (as a part of shapes.py)



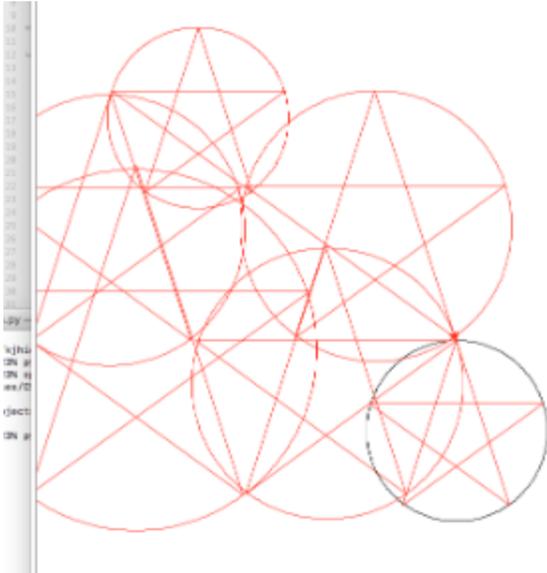
The third shape was simple in theory, though difficult in practice, as I had to get the two to be roughly the same size, and start the procedure for the star at precisely the right angle. This required several iterations of editing the code by one digit at a time, running it, and trying again and again. To achieve this, I used variables 'size' and 'length' to modify multiple components of the code at once.

### Task 4: shapeFour (as a part of shapes.py)



Ah. It turns out I did shapeFour as part of shapeThree, since I was using the variables to get my shape to line up correctly, so here are two examples of my work. Once I figured out the correct ratio of size to length, I input those numbers farther back in the code so I could reduce them both to the same variable of 'size.'

### Task 5: shapeFive (as a part of shapes.py)

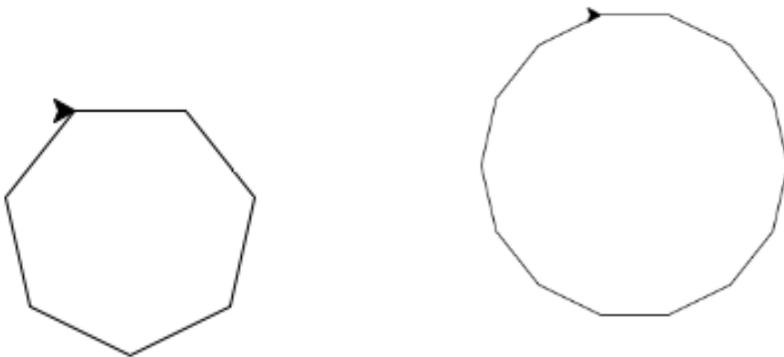


In order to accomplish shapeFive, I added several of them to the original shapeThree with different locations and sizes. As the turtle ends up pointing in another direction after every pentagram, I did not need to add any 'turn' commands as part of shapeFive. I simply used 'penup' and 'pendown' to create multiple shapes on the same line.

## Extensions:

### Extension 1:

*Nth sided shape*



I took a fair amount of time to discover the method required in order to make an Nth sided shape. I was sucked into the world of variables, and while describing the process necessary for making the internal angle was incredibly simple,  $(360/n)$ , simple math, it took me forever to discover how to make multiple commands, such as 'move' and 'turn' in order to complete the shape. Despondent, I turned to John Connors, a friend and dorm-buddy, as I knew he was knowledgeable in the ways of computers. He pointed me in the way of 'for loops,' and I found myself at the door of Hannah, begging for her help. She, in turn, aimed me towards online resources which I used to teach myself the ways of the loop. And so I made shapes.

## Questions:

1. What is a variable? Variables are stand-ins that connect the code to multiple spots in order to allow for one change to affect many lines of code.
2. What is a function and how are they useful? Functions are pre-set lines of code understood by python/other coding languages/apps that process information more effectively.
3. What is a parameter and how do they affect the generality of a function? Parameters are the input areas of functions, and they allow for functions to be used anywhere the function is applicable for different variables and values.
4. How can you use functions to create complexity with less code? You build functions upon functions - things that process pre-processed data in order to create more complex outputs with fewer inputs.
5. Pick a wall drawing by Sol LeWitt that you think is interesting. Why? Do you think you could write code for it in python? If I succeed in learning how to fill shapes, I believe I will be able to remake Wall drawing # 792 by Sol LeWitt using Python. I found it interesting, as these large black shapes are defined by slivers of white so much smaller than themselves; that these boundaries would lend such definition to the wall art.

## Conclusion:

This project, while fun, was a good intro to CS 151. It forced me to move carefully, think outside the box, and to ask for help. I learned how to draw using turtle, how to draw instantly, how to color the pen line, how to loop functions, how to use multiple variables, how to do basic math in python, and how this is going to be an interesting course.

## Acknowledgements:

John Conners, Hannah Wolfe, Bruce Maxwell, Bruce Maxwell's cookies, Sam Schultz

<https://docs.python.org/3/tutorial/controlflow.html#the-range-function>