Use the tools and resources from class and the sample function code below to help complete this assignment.

Sample Function 1: Positioning the Pen

The following function positions the turtle pen without drawing anything.   
The parameter “xPos” is the amount left or right to move the pen

The parameter “yPos” is the amount up or down to move the pen

*def movePen(xPos, yPos) :*

*myPen.up()*

*myPen.forward(xPos)*

*myPen.right(90)*

*myPen.forward(yPos)*

*myPen.left(90)*

*myPen.down()*

Sample Function 2: Drawing a Blue Box

The following function draws a blue box with the top left corner at the current pen position. The parameter “size” is the size of the box.

*def blueBox(valSize) :*

*myPen.down()*

*myPen.color( "blue" )*

*myPen.forward(valSize)*

*myPen.right(90)*

*myPen.forward(valSize)*

*myPen.right(90)*

*myPen.forward(valSize)*

*myPen.right(90)*

*myPen.forward(valSize)*

*myPen.up()*

Write your answers to the questions below in hard copy format. (i.e. pen or pencil)

1. Describe the shape produced by the following code:

*movePen(0,0)*

*blueBox(40)*

*movePen(20,60)*

*blueBox(40)*

*movePen(80,0)*

*blueBox(40)*

*movePen(-100, -60)*

*blueBox(40)*

This produces 4 hollow, blue squares in a shape similar to how the 4 main buttons on controllers are spaced and placed, where it draws one blue square, then draws another a little to the right and up relative to the first, then draws another except to the left and above the first box, as well as parallel to the second square, and finally draws a final blue square directly above (parallel) the first square and in the middle and above the second and third square creating an overall rhombus of the squares if you were to connect the middles of each square to each other.

1. Modify the blue box function to produce the following shape:

import turtle

myPen = turtle.Turtle()

def blueBox(valSize) :

myPen.down()

myPen.forward(valSize)

myPen.right(90)

myPen.forward(valSize)

myPen.right(90)

myPen.forward(valSize)

myPen.right(90)

myPen.forward(valSize)

myPen.begin\_fill()

myPen.right(135)

myPen.forward(valSize\*1.4)

myPen.right(135)

myPen.forward(valSize)

myPen.right(90)

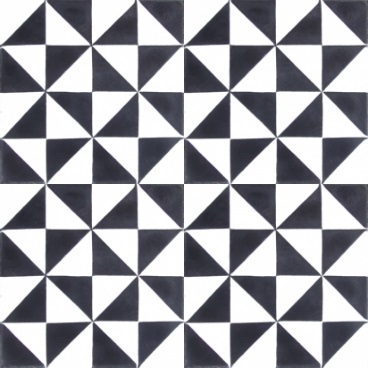
myPen.forward(valSize)

myPen.end\_fill()

myPen.up()

blueBox(50)

1. Write a program using your function(s) to produce a pattern similar to the following:



import turtle

myPen = turtle.Turtle()

myPen.delay(1)

def row1():

myPen.begin\_fill()

myPen.right(180)

myPen.forward(30)

myPen.right(135)

myPen.forward(30\*1.4)

myPen.end\_fill()

myPen.right(45)

myPen.begin\_fill()

myPen.forward(30)

myPen.right(90)

myPen.forward(30)

myPen.right(135)

myPen.forward(30\*1.4)

myPen.end\_fill()

myPen.right(135)

myPen.forward(30)

myPen.right(90)

myPen.forward(30)

myPen.left(90)

myPen.up()

myPen.forward(30)

myPen.down()

def row2():

myPen.begin\_fill()

myPen.forward(30)

myPen.left(90)

myPen.forward(30)

myPen.left(135)

myPen.forward(30\*1.4)

myPen.left(135)

myPen.forward(30)

myPen.end\_fill()

myPen.left(90)

myPen.forward(30)

myPen.right(90)

myPen.begin\_fill()

myPen.forward(30)

myPen.right(90)

myPen.forward(30)

myPen.right(135)

myPen.forward(30\*1.4)

myPen.end\_fill()

myPen.right(135)

myPen.forward(30)

myPen.right(90)

myPen.forward(30)

myPen.left(90)

rowCounter = 0

while (rowCounter < 4) :

row1()

row1()

row1()

row1()

myPen.up()

myPen.right(180)

myPen.forward(30)

myPen.down()

row2()

row2()

row2()

row2()

myPen.left(90)

myPen.up()

myPen.forward(60)

myPen.left(90)

myPen.forward(30)

rowCounter = rowCounter + 1