

Exercise 1 – Quilting in Python

*Source from The Sampler Quilt Exercise by Julie Zelenski 2001 at nifty.stanford.edu as well as Eric Roberts 1999

DATE DUE: Class 6

DATE ASSIGNED: Class 2

Goals:

This assignment will focus on the student becoming familiar with the basic programming concepts discussed in class and implemented in Python with Turtle graphics.

Requirements:

Using turtle graphics, produce a “sampler quilt” using turtle graphics and python. The quilt will consist of five blocks, each with its own theme. Each block is repeated five times and all are arranged to form the entire quilt.

Your quilt should include five unique patterns (suggestions - flowers, simple or complex geometric shapes – see suggestions below)

Start with an arrangement with each pattern in a column. 5 X 5

Next try arranging them in a table that is cyclic (modulus or “remainder operator %”) will come in handy here.

```
4 0 1 2 3 4 0
3 4 0 1 2 3 4
2 3 4 0 1 2 3
1 2 3 4 0 1 2
0 1 2 3 4 0 1
```

Define the quiltRows, quiltColumns, blockSize so that the user can easily change these along with the size of each block (assume the blocks are square)

[Recursion in a block is NOT required for this assignment and is only advised once you have a version of the quilt working without.]

Considerations:

You have worked on this exercise in class and will be allowing another student to try your result. Please use this as an opportunity for feedback for improvement. This is an exercise also in receiving and giving constructive criticism. This is also an exercise in going through the submission process of handing in exercises and projects.

Work incrementally – build one block at a time – strongly advised to use functions. Break this assignment down into individual pieces and test each individual part in isolation.

- Start early
- Think carefully about the overall program design – write up a plan – what functions, what arguments they will need
- Test your program in small pieces
- Write it so someone else is able to understand it easily

For advanced users, turn the sampler quilt into a screen saver.

Submissions guidelines:

Create a directory named LastnameFirstnameExercise1

- LastnameFirstname.py (You may submit multiple versions labeled LastnameFirstname.V1.py)
- a readme with any necessary instructions.

Grading:

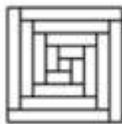
Clean efficient code and a cool visual are the emphasis.

Meeting the minimum specifications, 80%. To move your grade above 80% go beyond the specifications, demonstrate exploration and understanding. Consider using color, filled shapes, and so on – but first get the basic assignment working.

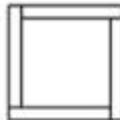
Be creative, have fun.

Suggestions for blocks:

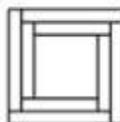
Log Cabin:



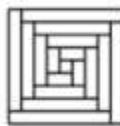
This block is a representation of an authentic quilt pattern that is widely used in traditional quilting. It is composed of a series of frames, each of which is nested inside the next larger one. Each frame is in turn composed of four rectangular boxes laid out to form a square. The outer frame, for example, looks like this:



The next frame fills the space inside the outer frame, as follows:



This same pattern continues for a specified number of frames, giving rise to the following complete picture:



numFrames = 4

frameWidth

The number of frames, let's call this numFrames and the overall size of the square, let's call this blockSize determine the frameWidth.

The frameWidth can be determined by the numFrames and the blockSize in the following manner:

$$\text{frameWidth} = \text{blockSize} / (2 * \text{numFrames} + 1)$$

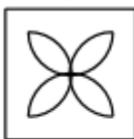
The frameWidth does not change in each frame, but remains constant for that given blockSize and numFrames. The equation is described by the fact that you need to determine how wide the rectangular strip needs to be so that each of the frames and the interior square will have the same width. If you look at the complete pattern you can see that each frame accounts for twice the frame width because there is a frame on each side. Since the square in the middle should be the size of that width, we divide the blockSize by twice the number of frames plus one for the square.

TIPS: To tackle this problem:

1. First draw a box – since we know we will be repeating this, define it as a function. (This is similar to the drawSquare function you did in class, however now you will specify a width and length in the parameters.
 2. Next, use that box function to draw a single frame (you will have to set the turtle's orientation appropriately for each box composing the frame.
 3. Make this a function as well, call it say drawFrame
- Next, determine what is different on each next level. You are drawing the same frame but now the overall size is smaller (the frameWidth is still the same). This will define your final function, drawLogCabin. Suggested parameters for this function are (t, x, y, blockSize, numFrames) where t is the turtle name and x and y are the position coordinates.

The solution will be discussed in class. Give it a try but don't get frustrated with this if you can't get it. It is not required to use this block pattern.

Flower Petals: The petal radius will be $.3 * \text{blockSize}$ (Solution for this is posted on the class website)

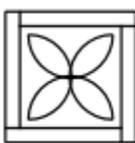


This block is a stylized representation of a four-petal flower and is included in the assignment mostly to give you practice drawing arcs. Each petal is composed of two quarter circles superimposed on top of each other as shown in the following enlarged picture of one petal:



The only constant required to draw the flower petal is the radius of the arc, which is defined as follows:

Hybrid:



This block is a combination of elements that appear in the log cabin (the outer frame) and flower petal designs. The point of including this block is to emphasize the importance of software reuse. If you've designed your program well, you will have to write very little new code to create this design.