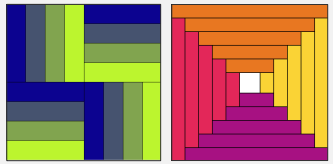
Lab 7

- choose convenient drawing modes (`CORNER` vs `CENTER`), animation variables, loop variables, function parameters
  - "convenient" means simpler expressions or even not needing an additional variable
  - e.g. `CORNER` mode for the rectangles in the rail fence and log cabin blocks is convenient because the left and top edges of the rectangles line up with the edges of the block



Lab 7

- "core concepts" topics assessed based on the presentation meeting and worksheet

Exercise 2

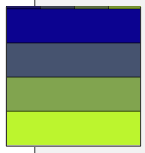
Start with the block as a whole – there are two copies each of the horizontal stripe pattern and the vertical stripe pattern. Two copies of something involving multiple shapes means a drawing function – answer the function questions below.

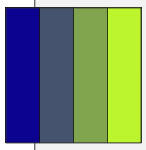
what is being drawn?	what differs from one copy to the next?
horizontal stripe pattern	
vertical stripe pattern	

these are the drawing function questions – this first step is about breaking down the rail fence block into two kinds of units (the horizontal stripe pattern and the vertical stripe pattern), with a drawing function for each kind of unit

"one copy to the next" is referring to the two copies of the horizontal stripe pattern and the two copies of the vertical stripe pattern in the rail fence block

this is not (yet) about how to draw the stripes themselves and what changes from one stripe to the next within each pattern

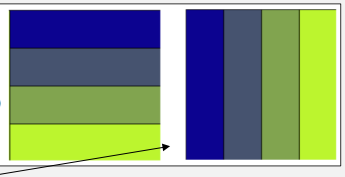




Lab 7

How is it drawn? Label the "what differs" things and other known quantities (such as the dimensions of each pattern) in the pictures.

(The loops to actually draw each set of stripes are addressed on the next page.)



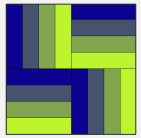
label the pictures!

start with labeling the "what differs" things identified (these will be the function parameters)

for example, if the position (x and y) is different for the two copies of the horizontal stripe pattern, label a point (x,y) in the horizontal stripe picture above – are you thinking of the position of the pattern as the upper left corner, the center, …? (pick something convenient)

then identify and work values or expressions for what you need to draw the shapes themselves

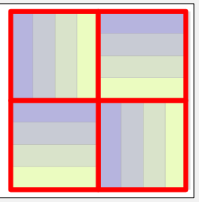
in this case, since there are loops involved in drawing the stripes, that will be tackled next – but for now, label any other known info about the pattern as a whole, such as the width and height of the region



Lab 7

What are the specific values for the “what differs” things?

Each red-outlined box shown to the right represents the pattern drawn by one use of the horizontal or vertical stripe functions. Label the picture with the specific values for each function call as well as which function (horizontal or vertical) is used to draw the pattern in each box.



label the picture!

when the functions are called, specific values will need to be passed for the parameters (the “what differs?” things)

for example, if you identified the upper left corner of each pattern as the position, label the upper left corner of each of the red-outlined blocks with the specific values or expressions for those positions when drawing the rail fence block in the entire drawing window

Lab 7

Getting back to the details of “how is it drawn?”, answer the loop questions separately for the horizontal and vertical stripe patterns. Also label specifically what changes in the pictures in those rows.

	horizontal stripe pattern	vertical stripe pattern
what is repeated?		
what changes from one repetition to the next?		
how do things start?		
how do things change?		
when do you keep going?		
is this a counting loop or a repeat-as-long-as pattern?		

now finish “how is it drawn?” by identifying the loop elements

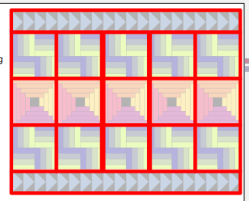
in addition to identifying in words what changes from one repetition to the next, label the pictures

for example, if you say “y changes”, label specifically what y refers to – is it the top of each rect, the middle, …? (pick something convenient for initializing, drawing, and/or updating)

also label the pictures with other info that helps you figure out the rest of the loop questions and what you’ll need to actually draw each rect – label what the function parameters refer to, the dimensions of the region, the dimensions of each rect, ...

Exercise 4

Each red-outlined boxes represents the pattern drawn by one use of the drawing functions you’ll create in this exercise.



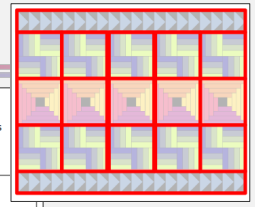
Answer the function questions below (and continued on the next page) for each of the drawing functions.

what is being drawn?	what differs from one copy to the next?
strip of flying geese	
single rail fence block	

these are drawing functions questions again, referring to drawing functions for each of the types of quilt blocks

“one copy to the next” is referring to the two copies of the flying geese strip, the 10 copies of the rail fence block, and the 5 copies of the log cabin block – the red-outlined boxes – *not* repetitions of shapes that form the patterns within each block

Lab 7



Now consider the entire quilt, and answer the loop questions below using the drawing functions as the building blocks.

There’s more than one way to break the pattern down into loop(s) so there are two columns below (for two loops) and another copy of the table on the next page – depending on what you decide on for your loops, you may not need all of the columns.

identify what part of the quilt is being drawn by the loop		
what is repeated?		
what changes from one repetition to the next?		
how do things start?		
how do things change?		
when do you keep going?		
is this a counting loop or a repeat-as-long-as pattern?		

these loop questions are about building the whole quilt from repeated blocks – the red-outlined units in the picture – *not* the repetition of rects and triangles within each block

“what is repeated?” should be “strip of flying geese”, “rail fence block”, “log cabin block”

Lab 7

- identifying the right loop pattern
  - pay attention to the words describing the pattern as well as looking at the picture

3. In this exercise you'll create a sketch which displays a single log cabin block, as shown. Pay careful attention to the example so you get details of the pattern correct!

The requirements for your sketch:

- Name the sketch **lab7c**.
- The drawing window should be square.
- Design your sketch so it works with any size window.
- The block should fill the entire drawing window, and each rectangle should be 20 pixels wide (vertical rectangles) or high (horizontal rectangles). While the real quilt block is built from the inside out, yours will be built outside in — include as many rectangles as fit, leaving a blank space in the middle if there isn't enough room to have another complete set of rectangles.

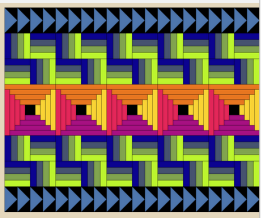
"as many rectangles as fit" → repeat as long as

(also, the number of rectangles changes with the size of the window rather than the rectangle size adjusting to ensure there are always 5)

Lab 7

- identifying the right loop pattern
  - pay attention to the words describing the pattern as well as looking at the picture

4. In this exercise you'll create a sketch which displays a quilt made from the blocks created in #1-3, as shown. The requirements for your sketch:



- Name the sketch **lab7d**.
- Choose a size for your drawing window so that the blocks fit nicely without distortion or extra space on one edge (in the case of the flying geese).
- Replicate the pattern shown: there should be a strip of flying geese at the top, then a row of 5 rail fence blocks, a row of 5 log cabin blocks, another row of 5 rail fence blocks, and a final strip of flying geese. (You can choose your own colors, but otherwise your quilt should match the example.)

"row of 5" → counting loop

(also, there's repetition here – use loops to draw the rows of rail fence and log cabin blocks instead of just pasting the function calls 5 or 10 times)

Lab 7

- include comments describing each function and its parameters
  - Include comments describing each function and its parameters. For example:

```
// draw a tree
//  (x,y) is the position of the bottom center of the tree trunk

void drawTree ( int x, int y ) {
  ...
}
```