1 Administrative Topics

- We take the quiz.
- Either delete the pages or remove the labels from pages you don’t mean to hand in. We use those labels for course organization and it is cluttering things :)
- Most you have put your code in Courses/CS151/. Many of you have put it in Courses/CS151/yourname/Proj1, but we would like it in Courses/CS151/yourname/private/Proj1

2 Advice for Project 2

2.1 How to fill

Turtle can’t fill multiple objects at once, and it can’t fill convex shapes, so if you want to fill something complicated, you need to fill each individual component. For example if you have a circle and a box next to each other, then you need to do something like this:

```python
turtle.begin_fill()
turtle.color('green')
for i in range(4):
    forward(100)
```
Also, the turtle doesn’t always do well if you turn fill on, then pick up the pen and move it, like this:

turtle.begin_fill()
turtle.up()
turtle.goto(100, 200)
turtle.down()
turtle.circle()
turtle.end_fill()

Instead, wait until you get the pen to the place you want the circle, then turn fill on.

2.2 Code organization

Don’t forget to comment your code. Every function should have a docstring explaining what it does. For any main code or function with a lot of lines in it, then use comments to explain the different parts of the code. For example, in my tree function, I use a comment to indicate which lines of code draw the trunk and which lines of code draw the leaves.

Simple shapes should have sizes. And it should be possible to draw them at any angle. We talked about a block in class.

Aggregate shapes should have scales. You can always set up your complex shapes so that they are always upright. We talked about a drawing of a house in class.

It is a good idea to design your scenes as if you are a child using simple shapes cut out of colored construction paper. Don’t try to fit perspective into it, unless you really want to.

To figure out what colors are legal, do a web search for rgb.txt. And follow whatever link you want to. The list of colors should be at least partly compatible with your Python installation.
We wrote code to draw a row of blocks, under the assumption that there is a block function already defined and that its parameters are x (x-position of bottom lefthand corner), y (y-position of bottom left-hand corner), and edge (length of edge).

```python
def rowOfBlocks( x, y, scale ):
    '''draw a row of blocks with the bottom lefthand corner at (x,y)
    If scale is 1, then each block will be 200x200.'''
    edge = 200
    block( x, y, edge*scale )
    block( x+1*edge*scale, y, edge*scale )
    block( x+2*edge*scale, y, edge*scale )
    block( x+3*edge*scale, y, edge*scale )
```

2.3 Summary guidelines

- Simple shapes have some sort of size associated with them (e.g. an edge length for a square)
- Aggregate shapes have a scale
- We scale offsets and sizes, but not positions.