My chess game will use functions (methods) inside classes for each piece type. The classes will be pawn, rook, knight, bishop, queen, king. They will have methods (functions) to output their string abbreviations and to get their location on the board. The parameters for this method consist of the piece number and color. I.e. (something like the following (written briefly in python syntax)):

class Rook(color, number):

def \_\_init\_\_(self):

self.color = color

self.num = number

def abbr(self):

if self.color == ‘black’: # only doing black for example

if self.num = 1:

return ‘br1’

else:

return ‘br2’

def get\_loc(self):

# still needs work

# might be easier to simply search the board for user input

#more code here

Return row,column of df

Now we can instantiate a rook piece like this:

black\_rook\_1 = Rook(black, 1)

Then access its abbreviation which will be printed in the 2D-array board like this:

black\_rook\_1.abbr()

This will happen for all 32 pieces (of 6 different types/classes).

I plan to declare the 6 classes as prototypes then instantiate the pieces in the main loop.