Name:_____ Section:___ Andrew Id: _____

15-112 Spring 2014 Quiz 6

* 15 minutes. No calculators, no notes, no books, no computers. No sets, maps, or recursion!

* Do not discuss this quiz with anyone until after 5pm today. SHOW YOUR WORK, CIRCLE YOUR ANSWERS.

1. **Code Tracing** [50 pts; 12.5 pts each]: Indicate what this will print, filling in the spaces provided on the right:

```
def ct1(a):
    (rows, cols) = (len(a), len(a[0]))
    result = [0]*rows
    for row in xrange(rows):
        for col in xrange(0,cols,2): # note the range!
            result[row] += a[rows-1-row][col] # note indices!
    return result
print ct1([[1,2,3,4,5],
          [2,4,6,8,10]])
```

_____

```
def ct2(x):
    result = [ ]
    a = range(x, x+2)
    for y in a:
        result.append([(y+z) for z in a])
    return result
print ct2(15)
```

_____

```
def ct3(n):
    result = [ ]
    for i in xrange(2,4):
        a = [ ]
        for j in xrange(i):
            a.append(n*i*j)
        result.append(a)
    return result
print ct3(3)
```

_____

```
def f(a):
    return 10*a[0][0]+a[1][0]

def ct4(a):
    b = copy.copy(a)
    c = copy.deepcopy(a)
    b[0][0] = 3
    c[0][0] = 4
    b[1] = [5]
    c[1] = [6]
    print f(b), f(c), # f is defined above
a = [[1],[2]]
ct4(a)
print f(a) # don't miss this
```

_____

Multiple Choice [50 pts; 7 pts each; 1 free pt]: Clearly circle your answers. “Tetris” == our specific implementation.

5. How did we modify the drawCell function so that it could be called both by drawBoard and by drawFallingPiece?
   a. We didn't. This is not how our code works.
   b. We added a "color" parameter.
   c. We added a boolean which is True when we are drawing the falling piece.
   d. None of the above.

6. Under what general conditions is the game over?
   a. The board is completely empty.
   b. The board is completely full.
   c. fallingPieceIsLegal() returns False immediately after placeFallingPiece() is called.
   d. The falling piece cannot fall any further.

7. Which function is most similar in structure to drawFallingPiece()?
   a. moveFallingPiece()
   b. newFallingPiece()
   c. fallingPieceIsLegal()
   d. rotateFallingPiece()

8. The function getCellBounds is a "model-to-view" function. Why didn't we also need a "view-to-model" function?
   a. We could also use getCellBounds as a "view-to-model" function.
   b. There is no need, since we do not use mouse input.
   c. We did implement "view-to-model" -- that's what newFallingPiece() does.
   d. None of the above.

9. Where do we use 3d lists?
   a. data.tetrisPieces
   b. data.tetrisPieceColors
   c. data.board
   d. None of the above

10. Why don't we first verify that a rotation is legal, rather than rotate, check if legal, and if not then unrotate?
    a. The rotate-check-unrotate approach is faster.
    b. The rotate-check-unrotate approach is an industry-standard way to implement Tetris.
    c. The rotate-check-unrotate approach shares more code with other piece movements.
    d. None of the above.

11. Which of the following will move a piece to the right? (be careful...)
    a. moveFallingPiece(data,-1,0)
    b. moveFallingPiece(data,+1,0)
    c. moveFallingPiece(data,0,-1)
    d. moveFallingPiece(data,0,+1)

12. **Bonus/Optional**: [2.5 pts]: Indicate what this will print:

```
a = [range(1,4),range(2,6),]
b = [r.pop(sum(r)/len(r)) for r in a]
print b, sorted(set(a[1])-set(a[0]))
```