15-112 Fall 2022 Lecture 3
Quiz 8A + 8B Handout

- **Use this handout to complete quiz8 (version A or version B).**
- **Submit this handout, but do not write on it.**

```
def testRectAndSquareClasses():
    print('Testing Rect and Square classes...', end='')
    assert(Rect.rectCount == 0)
    r1 = Rect(5, 10)
    assert(Rect.rectCount == 1) # increment with each Rect instance we create
    assert(r1.rectCount == 1)   # rectCount is also visible by each Rect instance
    assert((r1.width == 5) and (r1.height == 10))
    assert(str(r1) == '<5x10 Rect>')
    assert(str([r1]) == '[<5x10 Rect>]')
    assert(r1.getArea() == 50)

    r2 = Rect(5, 10)
    assert(Rect.rectCount == 2)
    assert(r1 == r2)
    r3 = Rect(10, 5)
    assert(str(r3) == '<10x5 Rect>')
    assert(r1 != r3)
    assert(r1 != 'Do not crash here!')

    s = set()
    s.add(r1)
    assert(r2 in s)
    assert(r3 not in s)

    # Square is a subclass of Rect
    n = Rect.rectCount
    square1 = Square(8)
    assert(isinstance(square1, Rect) == True)
    assert(Rect.rectCount == n+1)
    assert(not isinstance(r3, Square))
    # continued on next page!
```

```
assert(str([square1]) == '[<Square with side length of 8>]')
assert(square1.width == square1.height == 8)
assert(square1.getArea() == 64)

# makeTallRect is a method only for Squares, not Rects, and it returns
# a Rect with the same width as the square but twice the height:
assert('makeTallRect' not in dir(Rect)) # not defined in Rect
r4 = square1.makeTallRect()
assert(isinstance(r4, Rect))
assert(str(r4) == '<8x16 Rect>')

# getSquare returns a new Square instance that just fits in
# the given rectangle (so its size is the smaller of the width
# and height of the Rect):
r5 = Rect(7, 5)
square2 = r5.getSquare()
assert(isinstance(square2, Square))
assert(str(square2) == '<Square with side length of 5>')

# makeSquare does not return a new Square.  Instead, it mutates
# the rectangle so that it is square, so its new width and height are
# equal to the smaller of its current width and height.  Note that
# this does not turn the Rect object into a Square object.
r6 = Rect(7, 5)
assert(r6.makeSquare() == None) # the method returns None!
assert(isinstance(r6, Rect))
assert(str(r6) == '<5x5 Rect>')

# finally, a (silly) method (note how it is being called
# in two different ways here):
assert(Square(1).amazing() == 'yes!')
assert(Square.amazing() == 'yes!')

print('passed!')
```