4) What does the following code print? (10 points)

class Shape(object):
    def __eq__(s1, s2):
        return s1.area() == s2.area()
    def __lt__(s1, s2):
        return s1.circum() < s2.circum()

class Rectangle(Shape):
    def __init__(self, h, w):
        self.height = float(h)
        self.width = float(w)
    def circum(self):
        return 2*(self.height + self.width)
    def __str__(self):
        return 'Rectangle with area ' + str(self.height*self.width)

class Square(Rectangle):
    def __init__(self, s):
        Rectangle.__init__(self, s, s)
    def __str__(self):
        return 'Square with side ' + str(self.height)

class Circle(Shape):
    def __init__(self, radius):
        self.radius = float(radius)
    def circum(self):
        return 3.14159*(2*self.radius)
    def __lt__(self, other):
        return self.radius < other.radius
    def __str__(self):
        return 'Circle with diameter ' + str(2.0*self.radius)

def reorder(L):
    for e in L:
        if e < L[0]:
            L[0] = e

L = [Square(6), Rectangle(2, 3), Circle(1)]
try:
    reorder(L)
    for e in L:
        print e
except:
    for e in L:
        print e
7) Next to each item in the left column write the letter labeling the item in the right column that best matches the item in the left column. No item in the right column should be used more than once, and no box should contain more than one letter. (12 points)

- data abstraction     a) inheritance
- merge sort          b) divide and conquer
- polymorphism        c) $O(\log n)$
- hashing             d) $O(n)$
-                                                    e) $O(1)$
-                                                    f) specification
-                                                    g) mutability

8) Do you think that the lectures are too slow paced, too fast paced, about right?

   Too slow  1   2  3  4  5   Too fast

9) Do you think that the problem sets are too short, too long, about right?

   Too short  1   2  3  4  5   Too long