Name: ____________________ Recitation: ______ Andrew Id: ____________________

15-112 Fall 2017 Quiz 9

Up to 20 minutes. No calculators, no notes, no books, no computers. Show your work!

1. (20 points) **Short Answer**: Calculate the efficiency of loops A, B, C, and the entire function fn1 in Big-Oh notation, where n is the length of the list lst.

```
import string
def fn1(lst):
    # LOOP A
    for i in range(len(lst)):
        for j in range(i+1, len(lst)):
            if lst[i] == lst[j]:
                lst[j] = lst[j].upper()
    # LOOP B
    letters = []
    for letter in string.ascii_lowercase:
        if letter in lst:
            letters.append(str(lst.count(letter)))
    # LOOP C
    while "0" in letters:
        letters.remove("0")
    return len(letters)
```

A:

B:

C:

fn1:

2. (10 points) **Short Answer**: Briefly describe what the difference is between a class and an instance, and give a quick example to demonstrate your point.

3. (30 points) **Free Response**: Write a recursive function countChars which takes a string as input and returns a 2-element list where the first element is the number of lowercase characters in the string and the second element is the number of uppercase characters in the string. For example, countChars("abcDE!") == [3, 2]. Your code may not use any loops.

4. (40 points) **Free Response**: Write the class Door that passes the following test cases. You may not hardcode any test cases.

```
door1 = Door(True, False)
assert((door1.closed == True) and (door1.locked == False))
assert(str(door1) == "Closed and unlocked door")
assert(str(Door(False, False)) == "Open and unlocked door")
door1.turnKey()
assert(door1.locked == True)
assert(str(door1) == "Closed and locked door")
assert(door1 == Door(True, True))
assert(door1 != Door(False, True))
assert(door1 != "Closed and locked door") # don’t crash!
s = set()
s.add(Door(False, False))
assert(Door(False, False) in s)
```

YOU MAY CONTINUE WRITING OOP CODE ON THIS PAGE.

USE THIS PAGE FOR SCRAP WORK. WORK ON THIS PAGE WILL NOT BE GRADED.