Name: ____________________________ Section: _____ Andrew Id : ______________________

15-112 Summer-1 2017 Quiz 5

***Up to 50 minutes. No calculator, no notes, no books, no computers. Show your work!**

1. **Code Tracing [20 pts]:** Indicate what these print. Place your answers (and nothing else) in the boxes below the code.

```
class Rectangle(object):
    def __init__(self, length=1, width=1):
        self.length = length
        self.width = width

    def __eq__(self, other):
        return self.length*self.width == other.length*other.width

    def __hash__(self):
        return hash(self.length*self.width)

class Square(Rectangle):
    def __init__(self, side=1):
        super().__init__(side, side)


def ct1():
    rect1 = Rectangle(length=10, width=5)
    rect2 = Rectangle(length=5, width=10)
    square1 = Square(5)
    square2 = Square()
    print("T:", type(square1) == Rectangle, type(rect2) == Rectangle)
    print("E:", rect1 == rect2, square1 == square2)

    s = set()
    setCheck = 0
    if rect1 in s: setCheck += 1
    s.add(rect1)
    if rect2 in s: setCheck += 10
    print("setCheck:", setCheck)

ct1()
```

```
def ct2(n):
    if(n == 1): return "1"
    else: return ct2(n-1) + " " + h(n)

def h(n):
    if(n == 1): return "1"
    else: return h(n-1) + " " + str(n)

print(ct2(3))
```

2. **Free Response – sumOfOddDigits(n) [20 pt]:** Write the function sumOfOddDigits(n) that takes a possibly negative integer and returns the sum of the odd digits. So for example sumOfOddDigits(1233) returns 7 because the odd digits 3 + 3 + 1 add up to 7. You are not allowed to use any iterations (so no for or while loops here).

3. **Free Response - getPyFiles(path) [20 pts]:**

Write a function that takes a path as input and returns the number of the files in that folder that are .py files. This function should recursively go into all the folders in the given path and count the .py files in those subfolders.

You may want to use os.path.isdir(path) and os.path.listdir(path) and remember that the function takes in a path, such as "Desktop/Folder1". You may also want to take advantage of the string method .endswith()

4. **Free Response - Book and Library Classes [40 pts]:**

Write the Book and Library classes so the following test code works. You may not hardcode the specific test cases.

```
book1 = Book("Harry Potter", "J. K. Rowling", 2007)
assert(book1.title ==  "Harry Potter")
assert(book1.author == "J. K. Rowling")
assert(book1.year == 2007)
book2 = Book("To Kill a Mockingbird", "Harper Lee", 1960)
assert(book2.title ==  "To Kill a Mockingbird")
assert(book2.author == "Harper Lee")
assert(book2.year == 1960)
library1 = Library([book1, book2])
assert(library1.books == [book1, book2])
assert(library1.averageAge() ==  1983.5) # average of 2007 and 1960
library2 = Library([book1])
assert(library2.books == [book1])
assert(library2.averageAge() == 2007)
assert(not (library1  ==  library2))
library2.addBook(book2)
assert(library2.books == [book1, book2])
assert(library2.averageAge() == 1983.5)
assert(library1 == library2)
```