Advanced Python Exam 3

Spring 2017

1. Do all work on this exam, using backs of pages and the extra   
 pages for extra space for answers or scratch

2. Closed book, no notes or electronic devices of any kind

3. Print your name on each page of the exam

4. No speaking except to the TA or the Instructor

Name (Print legibly): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

USF ID #: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Leave the box below blank; your score will entered there

Score:

1. (a) [13 points] Complete the definition of the generator function **oddsquares()** that will generate the squares of all positive odd integers.

(b) [12 points] You are to write a decorator **map\_generator(f)** which turns a function **f** into a generator that produces f(1), f(2) , … . This should be a straightforward modification of your answer to (a).

3. [20 points] Fill in the missing parts of the definition of the memoize decorator

**def memoize(f):**

**memo = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**def helper(x):**

**return helper**

3. [15 points] Define a class **Person** with two visible attributes: a string **name** and integer **id**. **id** must be a read-only property. The methods should include the **\_\_init\_\_** method, the **\_\_str\_\_** and **\_\_repr\_\_** methods, and methods that will allow two **Person** objects to be tested for equality and for “less than”. For the latter two, the comparison should be based on the **Person**’s **name** and **id**, with the **name** checked before the **id**.

**class Person:**

(b) [15 points] Define subclass **Student** of **Person**. In addition to the inherited Person attributes, A **Student** object should have two additional attributes: **major** ( a string) and **class** , also a string. Use properties to ensure that the value for class must be one of the following: freshman, sophomore, junior, senior.

The class must have **\_\_init\_\_**, **\_\_str\_\_** and **\_\_repr\_\_** methods. Use must use inheritance where possible.

**class Student(Person):**

4. (a) [5 points] The immutable version of list is tuple. What type is the immutable version of set?

(b) [15 points] Using the set type, write a single line statement that will produce a list L of the unique values of an   
arbitrary list K.   
  
Example: if K = [ 8, 2, 3, 8, 2, 6, 3], then L could be [2, 8, 3, 6].

L = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(c) [5 points] Assuming the above, fill in the sorting key to rearrange the values in L so that they in the order they first

appear in K. In the above example, the sorted version of L would be [8, 2, 3, 6].

L.sort(key = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)