Week4

Week 4 Plan

1. review exercise, dictionary and list
2. object oriented programming (OOP) in python
3. simple exercise OOP (more exercise next week)

Revisit exercise 3-2

- (using dictionary to look up in another dictionary)

dictionary (revisited)

- unlike list, its a pair of key & values
- you can get the value when you specify the key! i.e you don’t need to specify index (position)

- example: mylist ={‘a’:1, 2:[‘golf’,1],(1,’p’,3):4, 1:3, 5:’b’}

- mylist[‘a’] will return 1
- mylist[2] will return [‘golf’,1]

- a list can only be used as value, not as key but a tuple can be used as key

Let's go through some basic concepts/terminology of OOP

- class, object/instance
- method
- instance variable, class variable
- inheritance

Read this!: http://www.python-course.eu/object_oriented_programming.php

Review the examples/files listed in the next page

- protected & private (encapsulation)

(*) polymorphism

Examples

Review these !(download code from SFS)

<Employee.py>

<Customer.py>

<oop.py>

<Parent.py>

<Child.py>

<str.py>

OOP(cont)

- “self” in def
- “object” in class
- Same file vs different file as class definition
-

Exercise 4-0

Bank Account

Class name: account

Variables: balance , interest rate

Methods:deposit, withdraw, calcinterest, dispbalance

Create a python class for the above! <bank.py>

Exercise 4-0 (cont)

Test the <bank.py> above by creating another python file <ex40.py> that use the class this way: (or you can use the same file). If you want to use <ex40.py> to call (bank.py> then you need to do an import this way: from bank import account

Create a bank account object from the class

Set balance to 0 yen;

Deposit 4000 yen ;

Print account balance

Withdraw 500 yen

Print account balance

Deposit 1000 yen

Set the number of year to 7 (x=7)

Calculate the balance after 7 years (gaining 5% interest, n=5)

Print account balance . Hints: $Balance=InitBalance(1+n)^x$

Exercise 4-1 and 4-2 (when time permits)