

**CIS 180 L - Intro to Python - Fall 2006
Homework Assignment #8**

Due: THURSDAY, October 26th, beginning of class

Purpose: To practice a bit with Python classes

Note: as long as you meet the specifications below, you may add additional embellishments as you wish.

How to turn this in: submit the file **hw8.py** using `~st10/180pysubmit` on cs-server.

1. Create a plain-text file **hw8.py**. Start it with a **docstring** containing at least a brief description of this module, your name, and the last modified date.

Write an **if** statement that will, if this module is called from the command-line, print out some statement exclaiming that it is being called from the command line. (It should **not** print this statement if it is being imported within **python** or **IDLE**.)

2. Now consider **lect09_6.py** and **lect09_7.py**. These declare **Employee** and **Chef** classes (where **Chef** is a subclass of **Employee**).

Add lines doing the following to your **hw8.py**:

- * import **lect09_6** and **lect09_7**. (It is your choice whether you use **import** or **from** for this.)
- * create at least one **Employee** object, and create at least one **Chef** object
- * print to the screen the result of **explicitly** calling at least one additional method of your **Employee** object, and at least one additional method of your **Chef** object.

(Why do I say "explicitly" and "additional" above? Because you have implicitly called those class's constructors already, to create your objects.)

- * recall that the **dir** function lists all of an object's attributes. Call **dir** twice, once for each of your **Employee** and **Chef** objects, to show their attributes (and to see how the **Chef** object really has inherited `giveRaise` from **Employee**)

(It would be ideal if you can run **hw7.py** from the command line as well as import it in **python**/**IDLE**, so you can see the difference in the result of your if-statement in problem #1... 8-)

Now you can submit **hw8.py** using `~st10/180pysubmit` on cs-server.

Because the time between Tuesday and Thursday is too short, I won't formally ask you to write another subclass of **Employee**, such as a **Programmer** which should also have a **faveLang** attribute, an overridden **work()** method which prints "`<lastname> programs`", a method **changeLang** which changes the **faveLang** of that programmer to the new string given as its argument, and an overridden **__repr__** giving an appropriate string for the **Programmer** instance --- but it *would* be good practice. Especially if you were to then practice creating a **Programmer** object, calling its methods, and then seeing what that **Programmer** object's attributes are...