**Homework #2 CS2021- September 2, 2015**

**Working with Files, Objects, and Persistent Storage – in Python.**

We will work through the Preview directory of the book’s PP4E source code -- a demo of python in action. Last week’s exercise we explored data representation, and now we will consider files, pickling, OOP, object persistence, GUIs, and website basics. We won’t study these in depth at this time, but we will return to each at later date

Here we will run several script files to manipulate database files and process data persistently.

To import the book’s entire package code in other programs, copy the PP4E directory to a working directory on your computer's file system, and add the directory immediately above PP4E (i.e. its container on your machine) to your PYTHONPATH setting, or add the containing directory to the sys.path as discussed in Lecture Notes for Week 2.

Here we begin simply by copying the Preview subdirectory from the PP4E book’s source code repository to a working directory. Note that the Preview directory is not a package, since it does not contain an \_\_init\_\_.py file

**Step 2: Storing Records Persistently**

Here we will be introduced to the standard pickle module for pickling data into files, and the more convenient shelve module.

Go to the command line and execute the following:

Preview> python make\_db\_file.py

Preview> python

>>> for line in open('people-file'):

... print(line, end='')

...

bob

job=>'dev'

pay=>30000

age=>42

name=>'Bob Smith' etc………..

***Load, examine,* and run** each of the following scripts (check off each one as you complete task):

* Example 1-3. dump\_db\_file.py // Basic DB load in
* Example 1-4. update\_db\_file.py //Updates a few of the entries
* Example 1-5. make\_db\_pickle.py // dumps it into a file
* Example 1-6. dump\_db\_pickle.py // loads file and shows it
* Example 1-7. update-db-pickle.py // open, dump new information, close
* Example 1-8. make\_db\_pickle\_recs.py // establishes keys
* Example 1-9. dump\_db\_pickle\_recs.py // each has own file
* Example 1-10. update\_db\_pickle\_recs.py // update, open, close
* Example 1-11. make\_db\_shelve.py // a shelf is a dictionary like object
* Example 1-12. dump\_db\_shelve.py // data base dictionary stuff off shelf
* Example 1-13. update\_db\_shelve.py // fetch update and add record

**Step 3: Object Oriented Programming**

* Example 1-14 person\_start.py simple class, instantiate .name
* Example 1-15. person.py // adding entry in class
* Example 1-16. manager.py // class with basic raise function
* Example 1-17. person\_alternative.py // much bigger class
* Example 1-18. make\_db\_classes.py // imports the stuff we just created
* Example 1-19. dump\_db\_classes.py // same thing but from shelf
* Example 1-20. update\_db\_classes.py // import shelf + for loop

**Step 4: Adding Console**

* Example 1-21. peopleinteract\_query.py takes in input
* Example 1-22. peopleinteract\_update.py same but with interaction

**Step 5: Adding a GUI**

* Example 1-23. tkinter001.py // small pop up gui ‘spam’
* Example 1-24. tkinter101.py // button press and confirmation
* Example 1-25. tkinter102.py // button press and confirmation
* Example 1-26. attachgui.py // main pop up
* Example 1-27. customizegui.py // you can change what goes in the pop up
* Example 1-28. tkinter103.py // take input and give back out for pop up
* Example 1-29. peoplegui.py // takes in all info based on key

**Step 6: Adding a Web Interface**

* Example 1-30. cgi101.html // couldn’t get to work properly, html file
* Example 1-31. cgi101.py // couldn’t get to work properly, html file
* Example 1-32. webserver.py // not sure this did much but asked for permis
* Example 1-33. peoplecgi.html // couldn’t get to run
* Example 1-34. peoplecgi.py // couldn’t get to run