Assignment 3 supplementary hints

Note that these hints comprise explicit steps to take, in case you're not sure how to structure your solution. You may want to take a crack at the solution be referring here.

Please note that any code examples here may not run properly, because they are largely incomplete. The purpose here is to offer tips and tools towards a solution, not the solution itself.

1: inherit from **dict**

class ConfigDict(dict):

Inheriting from **dict** means that any **ConfigDict** object will behave like a dictionary.

2: constructor steps: set the filename, check for existence

class ConfigDit(dict):

def \_\_init\_\_(self, filename):

# set the name of the file in the instance

# use a "private" attribute for this

# (i.e., one that won't be used externally)

# ("private" names start with a single underscore)

# check to see if the file exists

# if it does:

# open the file and read it, line-by-line

# for each line in the file

# strip the line

# split the line on a '='

# set the key and value in the instance:

# NOT self[key] = value

# dict.\_\_setitem\_\_(self, key, value)

3: checking to see if a file exists

import os

if os.path.isfile('filename.txt'):

4: splitting only once (if a config value also has an equals sign)

configvalue = 'this=that=other' # meant to be key: **this**

# value: **that=other**

key, val = configval.split('=', 1) # **maxsplit** argument (1)

5: when setting a key and a value in the object, mustn't use this form:

self[key] = value

because it will call your class' **\_\_setitem\_\_** method. instead. use the parent class:

dict.\_\_setitem\_\_(self, key, value)

6: **\_\_setitem\_\_** steps

def \_\_setitem\_\_(self, key, value):

# (called when user says thisdict[key] = value)

# set key, value in instance

# (remember, dont' use self[key] = value!)

# instead, call the dict() constructor -- as was done in \_\_init\_\_

# i.e. dict.\_\_setitem\_\_(self, key, value)

# open the file (use the name you stored in the instance)

# loop through instance's key / value pairs

# write each key/value on a line, joined by an equals sign

# fh.write('{0}={1}\n'.format(key, value))

As always, good luck!