HW1

Reading:

HW Guidelines (col > documents > admin)

Chapter 7 and Sections 8.1 and 8.2

Programming:

Submit a single file named hw1.py that contains the solutions to the two problems below. When you are finished, test your solutions using `doctest`. Include the following code at the bottom of your module:

```
if __name__=='__main__':
    import doctest
    print( doctest.testfile( 'hw1TEST.py'))
```

1. Develop a class `BankAccount` with six methods:

   a. `__init__` - constructor, is able to construct a bank account with a given balance, or, if no balance is specified the balance defaults to 0. (see runs below)
   b. `__repr__` - converts `BankAccount` to a `str` for display, see runs below.
   c. `set` which takes a numeric amount as a parameter and sets the account balance to that parameter
   d. `withdraw` - which takes a numeric amount as a parameter and withdraws the amount specified by the parameter from the balance on the account
   e. `deposit` - which takes a numeric amount as a parameter and deposits the amount specified by the parameter from the balance on the account
   f. `balance` - which takes no parameters and returns the current balance on the account

```
# constructor/repr

>>> b = BankAccount(100)
>>> b
BankAccount(100)
>>> b = BankAccount()
>>> b
BankAccount(0)
>>> [b] # check that str is returned, not printed
[BankAccount(0)]
```

```
# methods

>>> b.deposit(50.25)
>>> b
BankAccount(50.25)
>>> b.withdraw(17.50)
>>> b
BankAccount(32.75)
>>> b.balance()
32.75
>>> b.balance()==32.75 # check balance is returned, not
printed
True
```

2. Write a function `processAccount`() that takes one argument, the name of a file The first line of the file is a number, which indicates the initial balance for a bank account. The remaining lines consist of a single character followed by a space followed by a number. The character will be one of 'd', 'D', 'w', or 'W', indicating that the amount that follows is either a withdrawal ('w' or 'W') or a deposit ('d' or 'D'). The function will create a new `BankAccount` object and then process each line of the file by calling the appropriate method of the `BankAccount` object. As it processes each line, it will print the information about the transaction being processed. The first line will be a call to the `set`() method and the remaining lines will be calls either to the `deposit`() or `withdraw`() method. Once the file has been processed, the function will print the final account balance and return the `BankAccount` object.

```
>>> processAccount('acct1.txt')
BankAccount(605.5500000000001)
>>> b = processAccount('acct1.txt')
>>> b
BankAccount(605.5500000000001)
>>> b.balance()
605.5500000000001
>>>
>>> b = processAccount('acct2.txt')
>>> b
BankAccount(46.94)
>>> b.balance()
46.94
>>>
>>> b = processAccount('acct3.txt')
>>> b
BankAccount(216.64)
>>> b.balance()
216.64
>>>
```