Homework 3

In this homework, you will design and implement a class which consists of the attributes and methods, like what you may need for the Resource Management functionality (Hardware Set) in your team project. The approach that you take for this homework can also be extended for Project Management functionality as well.

Fig.1 below shows screenshot of the driver code that uses the class `HWSet` that you will be developing as part of this homework

```
#This is the driver code that uses the hardwareSet class that you are writing.
import hardwareSet
#Create object hwSet1 of class hardwareSet with capacity of 250
hwSet1=hardwareSet.HWSet(250)
#print initial capacity units of hardware set 1
print("Total capacity of units:", hwSet1.get_capacity())

#print number of available units of hardware set 1
print("Number of available units:", hwSet1.get_availability())

#Create a list of two test items
test=[20,300]
#Run the test for all items in the test list
for i in test:

    err=hwSet1.check_out(i)
    #if function returns error code 0, it means we were able to checkout requested number of units, else we were
    #not able to check out requested number of units
    if (err==0):
        #print number of units available after checkout
        print("Number of units available after checking out", i, "units:", hwSet1.get_availability())
        #print number of checkout units
        print("Number of total checkedout units", hwSet1.get_checkedout_qty())
    else:
        #print number of units available after checkout
        print("Number of units available after checking out", i, "units:", hwSet1.get_availability())
        #print number of checkout units
        print("Number of total checkedout units", hwSet1.get_checkedout_qty())
        print("Could not check out requested number of units")

#checkin 180 units
hwSet1.check_in(180)
#print number of units available after checkin
print("Number of units available after checking in 180 units:", hwSet1.get_availability())
```

Fig.1 : Screenshot of driver code

This program can also be found on Canvas under Files>>Homework.

Fig. 2 below shows the output of the driver code with correct implementation of `HWSet` class

```
Total capacity of units: 250
Number of available units: 250
Number of units available after checking out 20 units: 230
Number of total checkedout units 20
Number of units available after checking out 300 units: 0
Number of total checkedout units 250
Could not check out requested number of units
Number of units available after checking in 180 units: 180
```

Fig.2 : Screenshot of results after running driver code with correct implementation of HWSet class

For this homework, you are required to write a Python program that implements the class `HWSet` which initializes the class with the following attributes

`Capacity` --> total number of units. Initial value=qty
`availability` --> number of units available to check out. Initial value=Capacity

and the following methods
and the following methods
`__init__(self,qty)`

`get_availability()` --> accessor function to return the number of unused units

`get_capacity()` --> accessor function to return the total capacity of units

`get_checkedout_qty()` --> accessor function to return the total number of checkout quantities

`check_out(qty)` --> method that checks out number of units specified by qty. This method should update the number of units available after check_out. This method should handle the situation if the quantity requested is greater than the current availability in the following manner: Allow users to check out the number of units that are available and then return error =-1

`check_in(qty)` --> method that checks in number of units specified by qty. This method should update the number of units available after check_in. No error checking is required here

What to submit:

`hardwareSet.py` : A file that contains methods and attributes for the class HWSet

Rubric

	Points
`__init__`	1
`get_availability()`	1
`get_capacity()`	1
`check_out(qty)`	1

check_in(qty)	1
Get_checkedout_qty	1
Comments	1