|  |  |
| --- | --- |
| **class** Account:   **def** \_\_init\_\_(self, name, balance):  self.name = name  self.balance = balance  print(**"account created for "** + self.name)   **def** deposite(self, ammount):  **if** ammount>0:  self.balance += ammount   **def** withdraw(self, ammount):  **if** self.balance >= ammount:  **if** ammount>0:  self.balance -= ammount  **else**:  print(**"please enter a good amount"**)  **else**:  print(**"you dont have sufficient balance"**)   **def** showammount(self):  print(**"you have {}"** .format(self.balance))  **if** \_\_name\_\_ == **"\_\_main\_\_"**: *#use two underscores* soham=Account(**"Soham"**, 500)  soham.showammount()  soham.deposite(500)  soham.showammount()  soham.withdraw(10)  soham.showammount() | account created for Soham  you have 500  you have 1000  you have 990 |
| account created for Soham  you have 500  you have 1000  you dont have sufficient balance  you have 1000 |
| **from** datetime **import** datetime **class** Account:   **def** \_\_init\_\_(self, name, balance):  self.name = name  self.balance = balance  self.transection\_list=[]  print(**"account created for "** + self.name)   **def** deposite(self, ammount):  **if** ammount>0:  self.balance += ammount  self.transection\_list.append((datetime.today().time(), ammount))   **def** withdraw(self, ammount):  **if** self.balance >= ammount:  **if** ammount>0:  self.balance -= ammount  **else**:  print(**"please enter a good amount"**)  **else**:  print(**"you dont have sufficient balance"**)   **def** showammount(self):  print(**"you have {}"** .format(self.balance))   **def** show\_transaction(self):  **for** date, ammount **in** self.transection\_list:  **if** ammount>0:  transection\_type=**"deposite"  else**:  transection\_type=**"withdraw"** ammount \*= -1  print(**"money : {} , Transection type : {} , time : {}"** .format(ammount, transection\_type, date))  **if** \_\_name\_\_ == **"\_\_main\_\_"**: *#use two underscores* soham=Account(**"Soham"**, 500)  soham.showammount()  soham.deposite(400)  soham.showammount()  soham.withdraw(100)  soham.showammount()  soham.show\_transaction() | account created for Soham  you have 500  you have 900  you have 800  money : 400 , Transection type : deposite , time : 20:46:28.491151 |

STATIC METHOD

|  |  |
| --- | --- |
| **from** datetime **import** datetime **class** Account:   @staticmethod  **def** \_currentTime(): *#ASTATIC METHOD IS A METHOD THAT DOESN'T USES self PARAMETER* **return** datetime.today().now();   **def** \_\_init\_\_(self, name, balance):  self.name = name  self.balance = balance  self.transection\_list=[]  print(**"account created for "** + self.name)   **def** deposite(self, ammount):  **if** ammount>0:  self.balance += ammount  self.transection\_list.append((Account.\_currentTime(), ammount))   **def** withdraw(self, ammount):  **if** self.balance >= ammount:  **if** ammount>0:  self.balance -= ammount  self.transection\_list.append((Account.\_currentTime(), -ammount))  **else**:  print(**"please enter a good amount"**)  **else**:  print(**"you dont have sufficient balance"**)   **def** showammount(self):  print(**"you have {}"** .format(self.balance))   **def** show\_transaction(self):  **for** date, ammount **in** self.transection\_list:   **if** ammount>0: #if amount is POSITIVE then DEPOSITE  transection\_type=**"deposite"  else**:  transection\_type=**"withdraw"** #if amount is NEGATIVE then DEPOSITEammount \*= -1 #SINCE IT WAS -ve TO IDENTIFY, THEN WE GOTTA CHANGE THEM IN +ve   print(**"money : {} , Transection type : {} , time : {}"** .format(ammount, transection\_type, date))  **if** \_\_name\_\_ == **"\_\_main\_\_"**:   soham=Account(**"Soham"**, 500)  soham.showammount()  soham.deposite(400)  soham.showammount()  soham.withdraw(100)  soham.showammount()  soham.show\_transaction()    print(soham.\_currentTime()) #STATIC METHOD IS CALLED | account created for Soham  you have 500  you have 900  you have 800  money : 400 , Transection type : deposite , time : 2018-02-23 22:36:32.392405  money : 100 , Transection type : withdraw , time : 2018-02-23 22:36:32.392405  2018-02-23 22:36:32.392405 |

IT CREATES A PROBLEM;

* It provides the last DEPOSITE and WITHDRAWAL
* But always ignores 1ST TRANSECTION, that has been made at the time of INSTANCE OF THE METHOD CREATION

|  |  |
| --- | --- |
| print(**"\*"**\*50)  *#new entity* ivy=Account(**"Ivy"**, 900) ivy.showammount() ivy.deposite(500) ivy.showammount() ivy.withdraw(100) ivy.showammount() ivy.show\_transaction() | account created for Soham  you have 500  you have 900  you have 800  money : 400 , Transection type : deposite , time : 2018-02-23 23:07:01.380693  money : 100 , Transection type : withdraw , time : 2018-02-23 23:07:01.380693  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  account created for Ivy  you have 900  you have 1400  you have 1300  money : 500 , Transection type : deposite , time : 2018-02-23 23:07:01.380693  money : 100 , Transection type : withdraw , time : 2018-02-23 23:07:01.380693 |

SOLUTION

|  |  |
| --- | --- |
| **from** datetime **import** datetime **class** Account:   @staticmethod  **def** \_currentTime():  **return** datetime.today().now();   **def** \_\_init\_\_(self, name, balance):  self.name = name  self.balance = balance  self.transection\_list=[]  Account.transectionBeginning(self)  print(**"account created for "** + self.name)  **THIS WILL TAKE THE VALUE WHEN IT IS CALLED & INITIALIZES THE MOST RECENT PAYMENT**  **def** transectionBeginning(self): *#CREATING A METHOD, THAT WILL BE CALLED* **return** self.transection\_list.append((Account.\_currentTime(), self.balance))   **def** deposite(self, ammount):  **if** ammount>0:  self.balance += ammount  self.transection\_list.append((Account.\_currentTime(), ammount))   **def** withdraw(self, ammount):  **if** self.balance >= ammount:  **if** ammount>0:  self.balance -= ammount  self.transection\_list.append((Account.\_currentTime(), -ammount))  **else**:  print(**"please enter a good amount"**)  **else**:  print(**"you dont have sufficient balance"**)   **def** showammount(self):  print(**"you have {}"** .format(self.balance))   **def** show\_transaction(self):  **for** date, ammount **in** self.transection\_list:  **if** ammount>0:  transection\_type=**"deposite"  else**:  transection\_type=**"withdraw"** ammount \*= -1  print(**"money : {} , Transection type : {} , time : {}"** .format(ammount, transection\_type, date))  **if** \_\_name\_\_ == **"\_\_main\_\_"**:  soham=Account(**"Soham"**, 500)  soham.showammount()  soham.deposite(400)  soham.showammount()  soham.withdraw(100)  soham.showammount()  soham.show\_transaction() | account created for Soham  you have 500  you have 900  you have 800  money : 500 , Transection type : deposite , time : 2018-02-23 23:32:27.850205  money : 400 , Transection type : deposite , time : 2018-02-23 23:32:27.850205  money : 100 , Transection type : withdraw , time : 2018-02-23 23:32:27.850205 |

ANOTHER ISSUE :

Since the fields are public anyone can re-initialize it; that GIVES A PROBLEM, when <THE BOLDED LINES WILL NOT MATCH>

|  |  |
| --- | --- |
| **from** datetime **import** datetime **class** Account:   @staticmethod  **def** \_currentTime():  **return** datetime.today().now();   **def** \_\_init\_\_(self, name, balance):  self.name = name  self.balance = balance  self.transection\_list=[]  Account.transectionBeginning(self)  print(**"account created for "** + self.name)   **def** transectionBeginning(self): *#CREATING A METHOD, THAT WILL BE CALL* **return** self.transection\_list.append((Account.\_currentTime(), self.balance))   **def** deposite(self, ammount):  **if** ammount>0:  self.balance += ammount  self.transection\_list.append((Account.\_currentTime(), ammount))   **def** withdraw(self, ammount):  **if** self.balance >= ammount:  **if** ammount>0:  self.balance -= ammount  self.transection\_list.append((Account.\_currentTime(), -ammount))  **else**:  print(**"please enter a good amount"**)  **else**:  print(**"you dont have sufficient balance"**)   **def** showammount(self):  print(**"you have {}"** .format(self.balance))   **def** show\_transaction(self):  **for** date, ammount **in** self.transection\_list:  **if** ammount>0:  transection\_type=**"deposite"  else**:  transection\_type=**"withdraw"** ammount \*= -1  print(**"money : {} , Transection type : {} , time : {}"** .format(ammount, transection\_type, date))  **if** \_\_name\_\_ == **"\_\_main\_\_"**:  soham=Account(**"Soham"**, **501**) #1st initialization  **soham.balance=900** #re initializing it  soham.showammount()  soham.deposite(400)  soham.showammount()  soham.withdraw(100)  soham.showammount()  soham.show\_transaction() | account created for Soham  **you have 900**  you have 1300  you have 1200  **money : 501** , Transection type : deposite , time : 2018-02-23 23:45:08.679464  money : 400 , Transection type : deposite , time : 2018-02-23 23:45:08.679464  money : 100 , Transection type : withdraw , time : 2018-02-23 23:45:08.679464 |

ATTRIBUTES WHOSE NAME STARTS WITH SINGLE UNDERSCORE, is used for internal use only

REFACTORING THE balance ATTRIBUTE INTO \_balance : making it use only for internal uses,

|  |  |
| --- | --- |
| **from** datetime **import** datetime **class** Account:   @staticmethod  **def** \_currentTime():  **return** datetime.today().now();   **def** \_\_init\_\_(self, name, balance):  self.name = name  self.\_balance = balance #internal use  self.transection\_list=[]  Account.transectionBeginning(self)  print(**"account created for "** + self.name)   **def** transectionBeginning(self): *#CREATING A METHOD, THAT WILL BE CALL* **return** self.transection\_list.append((Account.\_currentTime(), self.\_balance))   **def** deposite(self, ammount):  **if** ammount>0:  self.\_balance += ammount  self.transection\_list.append((Account.\_currentTime(), ammount))   **def** withdraw(self, ammount):  **if** self.\_balance >= ammount:  **if** ammount>0:  self.\_balance -= ammount  self.transection\_list.append((Account.\_currentTime(), -ammount))  **else**:  print(**"please enter a good amount"**)  **else**:  print(**"you dont have sufficient balance"**)   **def** showammount(self):  print(**"you have {}"** .format(self.\_balance))   **def** show\_transaction(self):  **for** date, ammount **in** self.transection\_list:  **if** ammount>0:  transection\_type=**"deposite"  else**:  transection\_type=**"withdraw"** ammount \*= -1  print(**"money : {} , Transection type : {} , time : {}"** .format(ammount, transection\_type, date))  **if** \_\_name\_\_ == **"\_\_main\_\_"**:  soham=Account(**"Soham"**, 501)  soham.balance=900 #external use  soham.showammount()  soham.deposite(400)  soham.showammount()  soham.withdraw(100)  soham.showammount()  soham.show\_transaction() | account created for Soham  you have 501  you have 901  you have 801  #not making problems like before  money : 501 , Transection type : deposite , time : 2018-02-23 23:54:03.937892  money : 400 , Transection type : deposite , time : 2018-02-23 23:54:03.937892  money : 100 , Transection type : withdraw , time : 2018-02-23 23:54:03.937892 |