Python 1

Total Marks

10.0

Pass Marks

5.0

Marks Obtained

NA

Status

NA

Report

---

Start Time :  18 Dec 2020 09:13

|

End Time :  31 Mar 2023 00:00

Question 1 :

Create a class Account with the below attributes:  
int accntNo  
String accntName  
int accntBalance  
  
Create a constructor which takes all parameters in the above sequence.  
  
  
Create a class AccountDemo  
  
Create a default constructor in the AccountDemo class as below  
  
def \_\_init\_\_(self):  
pass  
  
Create a method depositAmnt which takes an Account object and amount to be deposited (amount) as input parameters. Update the balance i.e. Add the amount to the existing balance and return the updated balance  
  
Create another method withdrawAmnt which takes an Account object and amount to be deposited (amount) as input parameters. Deduct the amount from the balance and return the updated balance. Minimum balance to be maintained is 1000. i.e if the balance is becoming less than 1000 when deducting the withdrawal amount, the operation need to be stopped with a message "No Adequate balance"  
  
  
To test the code against your customized Input through console, the input data needs to be entered in the below order( as shown below in the sample input).  
  
The first three lines in the below sample input represents the input for three variables of account object i.e account no. (accntNo),account name (accntName) and account balance (accntBalance), with which the object will be created.  
  
The fourth line in the sample input is the input for the amount to be deposited in the account object and fifth line is the input for the amount to be withdrawn from the account object  
  
Sample input:-  
120  
Rajesh  
1500  
1200  
2000  
  
  
Sample output for the above input:-  
2700  
No Adequate balance  
  
Note:  
For more details on  
a. Input data entered from standard input  
b. How this data is processed  
c. The order of the input data  
  
please refer the below main program.  
  
Note:  
Please request you to use the below main program to test/run your code and submit this main along with your solution.  
Dont write separate main function again on your own.  
  
if \_\_name\_\_ == '\_\_main\_\_':  
acno=int(input())  
acname=raw\_input()  
acntbal=int(input())  
depamnt=int(input())  
withamnt=int(input())  
acnt=Account(acno,acname,acntbal)  
acntdemoobj=AccountDemo()  
print(acntdemoobj.depositAmnt(acnt,depamnt))  
print(acntdemoobj.withdrawAmnt(acnt,withamnt))

#Define the Account class here

class Account:

def \_\_init\_\_(self, accntNo, accntName, accntBalance):

self.accntNo = accntNo

self.accntName = accntName

self.accntBalance = accntBalance

#Define the AccountDemo class here

class AccountDemo:

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

pass

def depositAmnt(self, amt\_obj, amt):

amt\_obj.accntBalance+=amt

return amt\_obj.accntBalance

def withdrawAmnt(self, amt\_obj, amt):

if amt\_obj.accntBalance-amt >= 1000:

amt\_obj.accntBalance-=amt

return amt\_obj.accntBalance

else:

print("No Adequate balance")

return ""

#Sample main section.

#Do not remove the below portion of code.

if \_\_name\_\_ == '\_\_main\_\_':

acno=int(input())

acname=input()

acntbal=int(input())

depamnt=int(input())

withamnt=int(input())

acnt=Account(acno,acname,acntbal)

acntdemoobj=AccountDemo()

print(acntdemoobj.depositAmnt(acnt,depamnt))

print(acntdemoobj.withdrawAmnt(acnt,withamnt))