OOP LAB-1

Lab Statement

Authors: Anmol Agarwal, Jahanvi Shah

General Instructions:

1) Read the question carefully.
2) Indent your code to make it readable and easier to debug.

In this lab, you will be learning the basics of Java OOP concepts with the help of bank account. You will have to create and simulate the working of a bank account for a single user. You just have to complete the Class template given to you.

You are expected to create a class BankAccount having the attributes
balance, INTEREST_RATE, transactionCount, NUM_FREE, TRANS_FEE, MIN_BALANCE

Values to be assigned to the attributes are:
INTEREST_RATE=5%
transactionCount=0 (number of transactions performed)
NUM_FREE=3(number of free transactions)
TRANS_FEE=10 per each transaction above the number of free transactions (transaction fee for transactions after 3rd transaction)
MIN_BALANCE=1000

Description of the methods

1.)**getBalance** should return the current balance after the operations
2.)**getTransactionCount** should return the number of transactions carried out by the user
3.)**deposit** should return true if the amount you are trying to deposit is greater than 0. If it is less than 0 then should return false. If deposit is successful, i.e. the function returns true then increase the transaction count by 1

4.)**withdraw** should return true if the amount you want to withdraw + minimum balance is greater than your current balance, i.e. your account should always have the minimum balance.

Example: If your balance is 2000 and you want to withdraw 1500 then the function should return false. But if you want to withdraw 1000 then it should be successful.

5.)**addInterest** calculates the interest given the interest rate and balance and deposits that in the account. Update the balance after finding the interest.
interest=(INTEREST_RATE/100)*balance

Update the transaction count in both withdraw and deposit method only when they are successful. Make sure you deduct the transaction fee(TRANS_FEE) from the balance in both the methods if the number of transactions exceeds the number of free transactions(NUM_FREE)

Test Case No.	Method Name	Marks
1	getBalance()	1
2	getTransactionCount()	1
3	addInterest()	1
4	deposit()	1
5	deposit()	1
6	deposit()	1
7	addInterest()	1
8	withdraw()	1
9	withdraw()	1
10	withdraw()	1

Pointers to keep in mind:

1. Make sure to import java.io.* and java.util.*

2. Run it by creating a folder java and put class BankAccount.java. When you want to run it, type the following command in the terminal:

java -jar test.jar

(Remember to put test.jar in the same folder as ‘java’) Please type this out (don’t copy it from the PDF).