OBJECT ORIENTED PROGRAMMING

ONLINE 2

LAB STATEMENT

This lab aims at creating a management system for a Bank that enables it to keep a track of its various customer accounts and its finances.

ACCOUNT

A bank account is maintained by a bank to record the financial transactions between the customer/account holder and the bank. Accounts can be of several types. The types of accounts this bank offers are:

1) Fixed Deposit
2) Savings Account
3) Loans Account

(They are slightly unusual than the normal bank accounts)
Each bank account has a principal Amount which is the money borrowed (in case of loan) or the money deposited in the bank. Also each account has a unique account number associated with it, which also depends on the type of account opened in the bank. Finally it also requires a function to calculate the amount of returns that they would receive from the bank. Again the returns from the bank depend on the account type. For eg. in Fixed Deposits you have higher interest rates compared to savings accounts. Thus the Account Number and the returns from the account are dependent on the account type. Thus Account class needs to be an abstract class.
Details of each account type are as follows:

FIXED DEPOSITS

A fixed Deposit has a Principal amount,interest rate and the time period of the deposit. Account number of a fixed deposit is of the form **fdXXXX.** It starts with fd1001 for the first account and is allotted the account numbers fd1002,fd1003,... sequentially as more fixed deposit accounts are added.
The return of the fixed deposit is the principal amount plus the compound interest on the principal amount for the given time period.

SAVINGS ACCOUNT

Even a savings account has a principal amount,interest rate and the time period for the money deposited. Savings account enables the customer to deposit/withdraw money at their will. Account number of a savings account is of the form **savXXXX.** It starts with sav2001 for the first account and is allocated the account numbers sav2002,sav2003,..., sequentially as more saving accounts are added to the bank. It even shows the last 5 transactions in the form of string to the user. For eg. wwwdd means that 3 withdrawals are performed on the account followed by 2 deposits.
The return of the savings account is the principal amount plus the simple interest on the principal amount for the given time period.

LOANS ACCOUNT

A loans account has a Principal amount that is borrowed, interest rate and the time period for the payment of loan. Account number of a loans account is of the form **loanXXXX.** It starts with loan5001 for the first account and is allotted the account numbers loan5002,loan5003,... sequentially as more loan accounts are added to the bank.
The return of the loans account is the principal amount plus the compound interest on the principal amount for the given time period that is finally to be paid back to the bank.

IBANK

The bank needs to follow certain rules listed by the Reserve Bank of a country. The Reserve Bank provides a limit on the number of each type of account a bank cannot exceed. Thus it maintains the 3 constants namely MAX_LOANS,MAX_SAVINGS_ACCOUNT,MAX_FDS which denote the upper cap on the number of loans,savings account and fixed deposit accounts a bank can have. This is an interface that the bank class needs to implement in order to follow the rules set by the Reserve Bank.

BANK CLASS

Bank Class manages all the accounts abiding by the limits set by the Reserve Bank. It enables the bank to add/create a new account for a customer or even close/delete an account for a customer that wishes to discontinue the services from the bank, given its account number.The bank would even be interested in knowing the profit it earns through loans and the amount of interest it has to pay to its customers (through FDs and Savings Account).

Dependencies for Tests for Online 2:

AccountTest :

testReturnsandPower(): all The getReturns() Function in FixedDeposit, LoansAccount , SavingsAccount and static pow() function in Account Class.

FixedDepositTest :

testFixedDepositConstructor(): getAccountNumber() and getPrincipalAmount() in FixedDeposit Class and constructors of Account,FixedDeposit Class.

LoansAccountTest:

testLoansAccountConstructor(): getAccountNumber() and getPrincipalAmount() in LoansAccount Class and constructors of Account,LoansAccount Class.

SavingsAccountTest : all the tests need SavingsAccount constructor to be implemented

testSavingsAccountConstructor(): getAccountNumber() and getPrincipalAmount() in SavingsAccount Class and constructors of Account,SavingsAccount Class.

testTransactions(): getPrincipalAmount() , withdraw(), deposit() of .SavingsAccount class

testlastTransactions():getLastFiveTransactions(),deposit(),withdraw() of SavingsAccount class

BankTest :

all the tests need Bank,account,FixedDeposit,LoansAccount,SavingsAccount constructors implemented .

testConstructor(): test BankConstructor and IBank Interface

testCreateAccountandProfitFromLoan(): IBank, CreateAccount() and profitFromLoans().

testCreateAccountandIntresttoPay():IBank, CreateAccount() and interestToPay().

testdeleteAccount() : IBank, deleteAccount() and CreateAccount()