Lab 12
OOP – BSDS – Spring 2022

Task 01:

Create classes related to banking system (keep all data members private) and run driver program "task1". The details are as under:

- **Date: (**save in file date.py)

Data Members: day*, month, year*

Member Functions: *init, str*

Function Details:

str: Return string by concatenating day, month and year with colon as separator. Like *11:10:2022*

BankAccount: (parent class, save in file bankaccount.py, import date in this class)

Data Members: *account no, balance, account type, opening date, status*

Member Functions: *init, deposit, withdraw, credit, debit, get balance, str, print account, terminate, get status, get account type*

Class Members: *total accounts*

Function Details:

init: This function has two parameters balance (opening balance) and account type. Add one to total accounts and assign to account no. Assign account type and balance to relevant data member. Set active to status. Create object of *datatime* (for this *import datetime*, and create object by calling *now* function from *datetime* class). Create object of class Date by passing *day*, *month* and *year* to the *Date* object and assign to data member *opening date*. Finally, print message like this **`"Account no: 3 of FIXED type is opened with balance: 4067679"`**

deposit: This function has one parameters amount. Check status, if it active, **add** amount to balance and print message like "**`962 is debited into Account no: 3`**". If account is not active, print message **`"This account is terminated"`**

withdraw: This function has one parameters amount. Check status, if it active, and balance is greater than equal amount, **subtract** amount from balance and print confirmation message, otherwise, if short of balance, print message, how much amount is short. In case of status, print message **`"This account is terminated"`**

credit: This function is same as deposit; the only difference is that deposit is called by customer logically and credit is called by banking system to credit interest etc. in the account

debit: This function is similar to credit; the only difference is amount is **subtracted** from balance

get balance: Check if status is active return balance, otherwise print termination message

terminate: Set status to terminate

get status: Return status

get account type: Return account type

str: Check if status is active, return string concatenating account no, account type and balance, otherwise, return string concatenating account no and status only

print account: Print each data member with label like print (f'Account No: {self.acc...}\)

class level function get total accounts: return total accounts

CurrentAccount: (child class of *BankAccount* class, save in file currentaccount.py)

Data Members: No data member required

Member Functions: *init, debit_bank_charges*

Function Details:

init: This function has one parameter balance (opening balance). Call init function of super with balance and '*CURRENT*' (account type)

debit_bank_charges: This function has no parameter, call debit function of super class with value 100

SavingAccount: (child class of *BankAccount* class, save in file savingaccount.py)

Data Members: No data member required

Member Functions: *init, credit_daily_commision*

Function Details:

init: This function has one parameter balance (opening balance). Call init function of super with balance and '*SAVING*' (account type)

credit_daily_commision: This function has no parameter, just call credit function of super class with amount = balance x 0.008%

FixedDeposit: (child class of *BankAccount* class, save in file *fixeddeposit.py*)

Data Members: No data member required

Member Functions: *init, credit_commission*

Function Details:

init: This function has one parameter balance (opening balance). Call init function of super with balance and 'FIXED' (account type)

credit_commision: This function has no parameter, just call credit function of super class with amount = balance x 0.12%

withdraw: Override withdraw function of super class and print message **"This is a fixed deposit account, widthdrawal not allowed"**

Finally, run "Task1.py" and output should be like:

```
Account no: 1 of CURRENT type is opened with balance: 284419
Account no: 2 of SAVING type is opened with balance: 2057361
Account no: 3 of FIXED type is opened with balance: 4067679
3109130 is credited into Account no: 2
771 is credited into Account no: 3
464682 is credited into Account no: 1
41331.928 is credited into Account no: 2
962 is debited into Account no: 3
41662.583424000004 is credited into Account no: 2
170 is debited into Account no: 3
1809553 is debited into Account no: 2
4890657 is credited into Account no: 3
202949 is debited into Account no: 1
781 is debited into Account no: 3
4171929 is credited into Account no: 3
1575494.76 is credited into Account no: 3
677 is credited into Account no: 3
391 is debited into Account no: 3
494 is credited into Account no: 1
2972458 is credited into Account no: 1
897 is credited into Account no: 3
644 is debited into Account no: 3
376 is credited into Account no: 1
1 CURRENT 3519480
2 SAVING 3439932.5114240004
3 FIXED 14705156.76
Account No: 1
Opening Date: 6:11:2022
Type: CURRENT
Balance: 3519480
Account No: 2
Opening Date: 6:11:2022
Type: SAVING
Balance: 3439932.5114240004
Account No: 3
Opening Date: 6:11:2022
Type: FIXED
Balance: 14705156.76
```

Task 02:

Add more classes in previous task and run driver program "task2". The details are as under:

SuperSavingAccount: (child class of SavingAccount, save in file supersavingaccount.py)

Member Functions: *init, print_account, credit daily commission*

Class Members: Define class level constant member *MINIMUM_BALANCE* initialized with value **50000**

Function Details:

init: This function has one data member balance. Call init function of super with balance and balance and 'S_SAVING' (account type).

withdraw: Override withdraw function of super class. This account requires minimum balance (**50000** already discussed). Therefore, check if balance - amount is more than equal minimum balance, deducted amount from balance and print withdraw message, otherwise print relevant message.

print_account: Override print_account function of super class. Call same function from super class. In function add print statement to print minimum balance required here.

credit daily commission: Override super class function. Use interest rate 0.01

OverdraftAccount: (child class of CurrentAccount, save in file overdraft.py)

Member Functions: *init, debit bank charges, withdraw*

Class Members: Define class level constant member *MINIMUM_BALANCE* initialized with value -**50000 (** means, customer can overdraft of Rs. 50000/- with extra bank charges.

Function Details:

init: This function has one data member balance. Call init function of super with balance and balance and 'OD' (account type).

withdraw: Override withdraw function of super class. This account requires minimum balance (-**50000** already discussed). Therefore, check if balance - amount is more than equal minimum balance, deducted amount from balance and print withdraw message, otherwise print relevant message.

debit bank charges: Override debit bank charges function of super class and charge Rs. 500 as bank charges instead of Rs. 100.

Finally, run "Task2.py" and your output should be like:

```
Account no: 1 of CURRENT type is opened with balance: 523705
Account no: 2 of SAVING type is opened with balance: 1864062
Account no: 3 of FIXED type is opened with balance: 4253817
Account no: 4 of S_SAVING type is opened with balance: 4514993
Account no: 5 of OD type is opened with balance: 2055283
660 is credited into Account no: 4
4176050 is credited into Account no: 1
1703020 is debited into Account no: 1
14912.496000000001 is credited into Account no: 2
500 is debited into Account no: 5
100 is debited into Account no: 1
511 is debited into Account no: 1
100 is debited into Account no: 1
385 is debited into Account no: 4
510458.04 is credited into Account no: 3
45152.68 is credited into Account no: 4
418 is debited into Account no: 4
2939023 is credited into Account no: 5
710 is debited into Account no: 1
This account is terminated
1447301 is debited into Account no: 1
130 is debited into Account no: 1
1 False
2 False
3 FIXED 4764275.04
```

```
4 S_SAVING 4560002.68
5 False
Account No: 1
Opening Date: 6:11:2022
Type: CURRENT
Balance: 1547883
This account is terminated
Account No: 2
Opening Date: 6:11:2022
Type: SAVING
Balance: 1878974.496
This account is terminated
Account No: 3
Opening Date: 6:11:2022
Type: FIXED
Balance: 4764275.04
Account No: 4
Opening Date: 6:11:2022
Type: S_SAVING
Balance: 4560002.68
Minimum Balance Required: 50000
Account No: 5
Opening Date: 6:11:2022
Type: OD
Balance: 4993806
This account is terminated
```