Bank example:

```
class HDFCBankAccount:
    def __init__(self, user_name, main_bal, pin=1234):
        self.name = user_name
        self.__MainBal = main_bal
        self.__pin = pin
        self.__account_Acrive_Status = True
        self.__isAtmCardHolder = False
        self.__isCheckBookHolder = False
        self.__isAtmCardGotFreezed = False

    def __verifyPin(self, pin):
        return self.__pin == pin

    def __updateMainBalByWithDraw(self, amount):
        self.__MainBal -= amount
        print(f"You have withdrawn {amount} successfully. Main balance: {self.__MainBal}")

    def __updateMainBalByDeposit(self, amount):
        self.__MainBal += amount
        print(f"You have credited {amount} successfully. Main balance: {self.__MainBal}")

    def __show_MainBal(self):
        print(f"Main balance: {self.__MainBal}")

    def __raiseAtmCard(self):
        self.__isAtmCardHolder = True
        return f"ATM card approved"

    def __raiseCheckBook(self):
```

```
    self.__isCheckBookHolder = True
    return f"Check book approved"

def __ATMCardFreezing(self):
    self.__isAtmCardGotFreezed = True
    return f"ATM card got freezed"

def withdraw(self, amount, pin):
    if self.__account_Acrive_Status:
        if self.__verifyPin(pin):
            if amount > self.__MainBal:
                print("You are trying to withdraw more than your balance")
            else:
                self.__updateMainBalByWithDraw(amount)
        else:
            print("Incorrect PIN...")
    else:
        print("Your account is not active")

def deposit(self, amount, pin):
    if self.__account_Acrive_Status:
        if self.__verifyPin(pin):
            self.__updateMainBalByDeposit(amount)
        else:
            print("Incorrect PIN")
    else:
        print("Your account is not active")

def check_Bal(self, pin):
    if self.__account_Acrive_Status:
```

```
            if self.__verifyPin(pin):
                self.__show_MainBal()
            else:
                print("Incorrect PIN")

    def request_for_ATMCard(self):
        statusOfAtmCardApproval = self.__raiseAtmCard()
        print(statusOfAtmCardApproval)

    def request_for_CheckBook(self):
        statusOfCheckBookApproval = self.__raiseCheckBook()
        print(statusOfCheckBookApproval)

    def request_for_ATMCardFreezing(self):
        print(self.__ATMCardFreezing())

    def request_for_AccountFreezing(self):
        self.__account_Acrive_Status = False
        print("Your account has been freezed")


class SavingAccount(HDFCBankAccount):
    def __init__(self, name, bal):
        self.loanLimit = 300000
        super().__init__(name, bal)

    def personalLoanRaise(self, amount):
        if amount > self.loanLimit:
            print("You are exceeding your loan limit amount")
        else:
```

```
            print("Your loan request is within the limit. Loan will be granted soon")


class BusinessAccount(HDFCBankAccount):
    def __init__(self, name, bal):
        self.loanLimit = 2000000
        super().__init__(name, bal)

    def businessLoanRaise(self, amount):
        if amount > self.loanLimit:
            print("You are exceeding your loan limit amount")
        else:
            print("Your loan request is within the limit. Loan will be granted soon")


# Object creation
sAccount = SavingAccount("Vamsi", 50000)
bAccount = BusinessAccount("Rakesh", 1000000)

# Example use:
amount = int(input("Enter amount :- "))
bAccount.businessLoanRaise(amount)

o/p:
Enter amount :- 2300
Your loan request is within the limit. Loan will be granted soon
Or
class HDFCBankAccount:
    def __init__(self, user_name, main_bal, pin=1234):
        self.name = user_name
```

```
    self.__MainBal = main_bal
    self.__pin = pin
    self.__account_Active_Status = True
    self.__isAtmCardHolder = False
    self.__isCheckBookHolder = False
    self.__isAtmCardGotFreezed = False

def __verifyPin(self, pin):
    return self.__pin == pin

def __updateMainBalByWithDraw(self, amount):
    self.__MainBal -= amount
    print(f"You have withdrawn {amount}. Available Balance: {self.__MainBal}")

def __updateMainBalByDeposit(self, amount):
    self.__MainBal += amount
    print(f" You have deposited {amount}. Available Balance: {self.__MainBal}")

def __show_MainBal(self):
    print(f" Available Balance: {self.__MainBal}")

def withdraw(self, amount, pin):
    if self.__account_Active_Status:
        if self.__verifyPin(pin):
            if amount > self.__MainBal:
                print("Insufficient Balance")
            else:
                self.__updateMainBalByWithDraw(amount)
        else:
            print("Incorrect PIN")
```

```
        else:
            print("Account not active")

    def deposit(self, amount, pin):
        if self.__account_Active_Status:
            if self.__verifyPin(pin):
                self.__updateMainBalByDeposit(amount)
            else:
                print(" Incorrect PIN")
        else:
            print("Account not active")

    def check_Bal(self, pin):
        if self.__account_Active_Status:
            if self.__verifyPin(pin):
                self.__show_MainBal()
            else:
                print("Incorrect PIN")

class SavingAccount(HDFCBankAccount):
    def __init__(self, name, bal):
        self.loanLimit = 300000
        super().__init__(name, bal)

    def personalLoanRaise(self, amount):
        if amount > self.loanLimit:
            print(" Loan request exceeds limit")
        else:
            print("Loan request approved within limit")
```

```
class BusinessAccount(HDFCBankAccount):
    def __init__(self, name, bal):
        self.loanLimit = 2000000
        super().__init__(name, bal)

    def businessLoanRaise(self, amount):
        if amount > self.loanLimit:
            print(" Loan request exceeds limit")
        else:
            print("Loan request approved within limit")
#Creating accounts
sAccount = SavingAccount("Vamsi", 50000)
bAccount = BusinessAccount("Rakesh", 1000000)

print("\n--- Saving Account Transactions ---")
sAccount.check_Bal(1234)      # Check balance
sAccount.deposit(10000, 1234) # Deposit
sAccount.withdraw(2000, 1234) # Withdraw
sAccount.check_Bal(1234)      # Final balance

print("\n--- Business Account Transactions ---")
bAccount.check_Bal(1234)
bAccount.businessLoanRaise(500000)  # Loan request within limit
bAccount.deposit(20000, 1234)
bAccount.withdraw(50000, 1234)
bAccount.check_Bal(1234)
```

o/p:

```
--- Saving Account Transactions ---
 Available Balance: 50000
 You have deposited 10000. Available Balance: 60000
```

You have withdrawn 2000. Available Balance: 58000

Available Balance: 58000

-- Business Account Transactions ---

Available Balance: 1000000

Loan request approved within limit

You have deposited 20000. Available Balance: 1020000

You have withdrawn 50000. Available Balance: 970000

Available Balance: 970000