Date : 24-05-23

Experiment No.5

Aim : Write a program to implement multiple inheritance. Consider a class BankAccount with data members as account number, aadhar number, owner name, ROI and balance with member functions openAccount(), deposit(amount),closeAccount() and updateInterest(). Create an interface Debitable which has method withdraw(). Derive a class FixedDepositAccount from BankAccount having data member lockInPeriod. Override methods updateInterest() to update Simple Interest, and method closeAccount() to charge 5 % for closure of FD Account before lockInPeriod. Derive a class SavingAccount from class BankAccount and interface Debitable. The account numbers should be serial numbers of 5 digit and automatically assigned on object creation, such that all FDAccounts start with 55 and all Saving account start with 11.

[ROI for Saving Account is 4% and for FD – 1-2yrs-6% ; 2-5yrs-6.5% ; &gt;5yrs- 7%]

Code :

BankAccount.java

```
public abstract class BankAccount {
    int accountNo;
    String adharNo;
    String oname;
    float roi;
    double balance;

    BankAccount(String adharNo, String oname, double balance) {

        this.openAccount();
        this.adharNo = adharNo;
        this.oname = oname;
        this.balance = balance;
    }

    BankAccount() {
    }

    void openAccount() {
        if (this instanceof SavingAccount) {
            accountNo = AccountNoGenerator.getSavingAccount();
        } else if (this instanceof FixedDepositAccount) {
            accountNo = AccountNoGenerator.getFdAccount();
```

Date : 24-05-23

```
        }
    }

    public String toString() {

        return ("\n===================== Account Details
=====================\nAccount Number : " + accountNo + "\nOwner Name : " +
oname + "\nAdhar Number : " + adharNo
                + "\nROI : " + roi + "\nAccount Balance  : " + balance  +
"\n");
    }

    abstract void deposit(double amount);

    abstract void updateInterest();

    abstract void closeAccount();
}
```

SavingAccount.java

```
public class SavingAccount extends BankAccount implements Debitable {

    SavingAccount() {
    }

    SavingAccount(String adharNo, String oname, double balance) {
        super(adharNo, oname, balance);
    }

    public String toString() {
        return (super.toString());
    }

    void deposit(double amount) {
        this.balance += amount;
        System.out.println(
                "Your account has been credited with " + amount + "\nYour
updated Account Balance is : " + this.balance);
    }

    void updateInterest() {
        this.roi = 4;
        balance += ((balance * roi) / 100);
        System.out.println("balance After adding interest : " + balance);
    }

    void closeAccount() {
```

Date : 24-05-23

```
        System.out.println("Account closed\nAmount Refunded : " + balance);
        balance -= balance;
        System.out.println("Available Balance : " + balance);
    }

    public void withdraw(double amount) {

        if (amount > balance) {
            System.out.println("Error on withdraw!");
            return;
        }
        this.balance -= amount;
        System.out.println(amount + " has been debited from your account
\nYour Account Balance is : " + this.balance);
    }
}
```

FixedDepositAccount.java

```
class FixedDepositAccount extends BankAccount implements Debitable {
    int lockInperiod;

    FixedDepositAccount(String adharNo, String oname,double balance, int
lockInperiod) {

        super(adharNo, oname, balance);
        this.lockInperiod = lockInperiod;
    }

    public String toString() {
        return (super.toString());
    }

    void deposit(double amount) {
        System.out.println("you are not allowed to deposit");
    }

    void updateInterest() {
        if (lockInperiod >= 1 && lockInperiod <= 2) {
            roi = 6;
        } else if (lockInperiod > 2 && lockInperiod <= 5) {
            roi = (float) 6.5;
        } else if (lockInperiod > 5) {
            roi = 7;
        }
        System.out.println("Interest : " + ((balance * lockInperiod * roi) /
100));
```

Date : 24-05-23

```
        balance += ((balance * lockInperiod * roi) / 100);
        System.out.println("balance After adding interest : " +
balance);
    }

    void closeAccount() {
        int n = 6;
        if (n < lockInperiod) {
            System.out.println("====================================\nAccount
Closer : before lock in period!\nPenalty : " + (5 * balance / 100));
            balance -= 5 * balance / 100;
            System.out.println("Amount refunded : " + balance);
        } else{
            System.out.println("==================================\nAccount
Closer : After lock in period ");
            this.updateInterest();
            System.out.println("Therefore Amount refunded : " + balance);
        }

    }
    public void withdraw(double amount) {
        System.out.println("YOU ARE NOT ALLOWED TO WITHDRAW MONEY");
    }
}
```

Debitable.java(interface Debitable)

```
public interface Debitable {

    void withdraw(double amount);
}
```

AccountNumberGenerator.java

```
class AccountNoGenerator {
    static int number = 54999;
    static int number2 = 10999;

    static int getFdAccount() {
        number++;
        return number;
    }
    static int getSavingAccount() {
        number2++;
        return number2;
    }
}
```

Date : 24-05-23

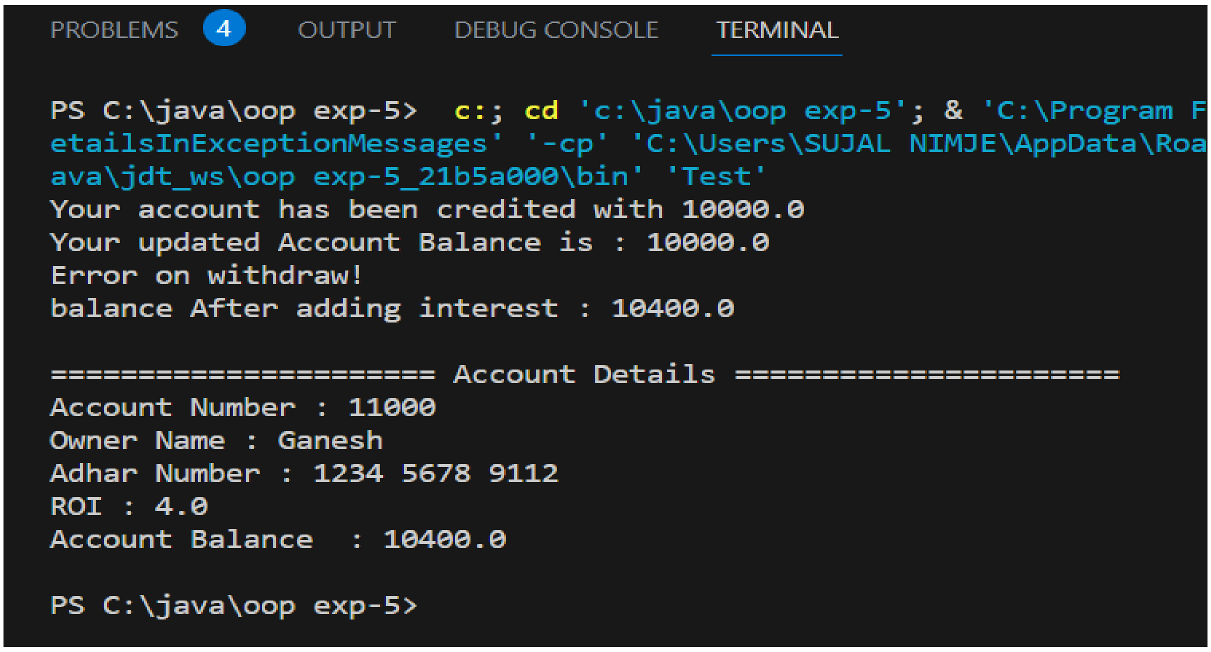
Test.java(main method)

```
public class Test {
    public static void main(String[] args) {

        SavingAccount s1 = new SavingAccount("1234 5678 9112", "Ganesh", 0);
        SavingAccount s2 = new SavingAccount("1234 5678 9000", "Bunti", 0);
        FixedDepositAccount f1 = new FixedDepositAccount("1111 5678 9000",
"Bubli", 200000, 5);
        FixedDepositAccount f2 = new FixedDepositAccount("1111 2222 9000",
"Vinayak", 200000, 5);

        s1.deposit(10000);
        s1.withdraw(20000);
        s1.updateInterest();
        System.out.println(s1);

        s2.deposit(50000);
        s2.withdraw(20000);
        s2.updateInterest();
        System.out.println(s2);
        s2.closeAccount();

        f1.deposit(50000);
        f1.withdraw(20000);
        System.out.println(f1);
        f1.closeAccount();

        f2.deposit(50000);
        f2.withdraw(20000);
        f2.closeAccount();
        System.out.println(f2);

    }
}
```

Date : 24-05-23

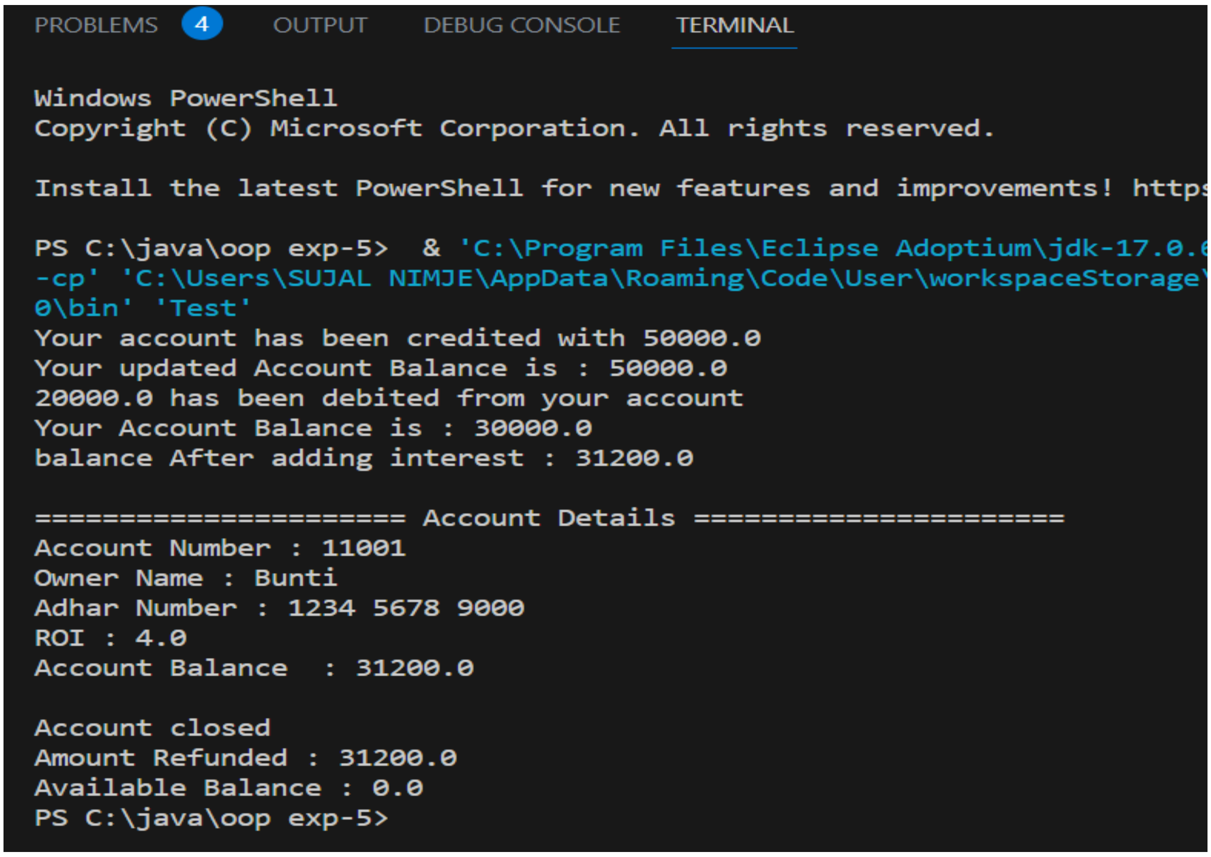
Output :

1. *Create a Savings Account Deposit Rs. 10000 to the account Withdraw 20000 from the account Add 1 year interest to the account Display account details*



```
PS C:\java\oop exp-5>  c:; cd 'c:\java\oop exp-5'; & 'C:\Program F
etailsInExceptionMessages' '-cp' 'C:\Users\SUJAL NIMJE\AppData\Roa
ava\jdt_ws\oop exp-5_21b5a000\bin' 'Test'
Your account has been credited with 10000.0
Your updated Account Balance is : 10000.0
Error on withdraw!
balance After adding interest : 10400.0

====================== Account Details ======================
Account Number : 11000
Owner Name : Ganesh
Adhar Number : 1234 5678 9112
ROI : 4.0
Account Balance  : 10400.0

PS C:\java\oop exp-5>
```

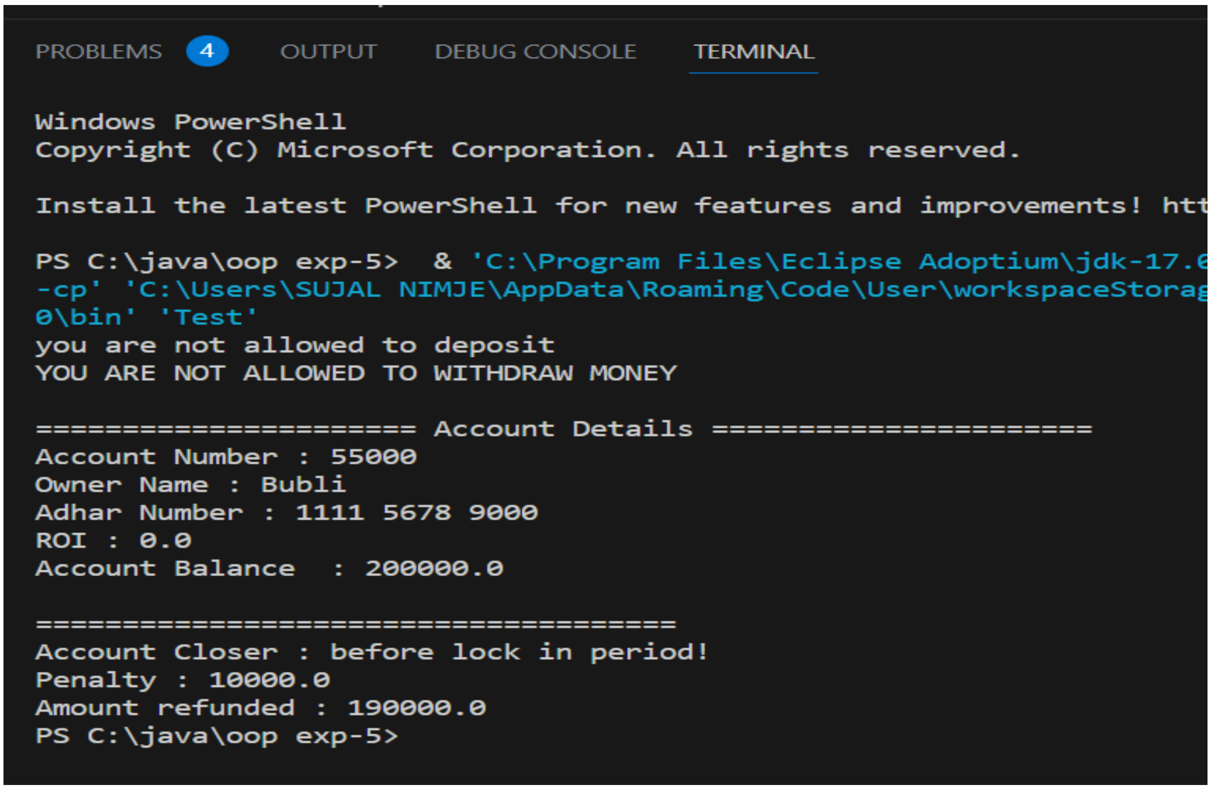
2. *Create a Savings Account Deposit Rs. 50000 to the account Withdraw 20000 from the account Add 1 year interest to the account Display account details Close the account*



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https

PS C:\java\oop exp-5>  & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.
-cp' 'C:\Users\SUJAL NIMJE\AppData\Roaming\Code\User\workspaceStorage
0\bin' 'Test'
Your account has been credited with 50000.0
Your updated Account Balance is : 50000.0
20000.0 has been debited from your account
Your Account Balance is : 30000.0
balance After adding interest : 31200.0

====================== Account Details ======================
Account Number : 11001
Owner Name : Bunti
Adhar Number : 1234 5678 9000
ROI : 4.0
Account Balance  : 31200.0

Account closed
Amount Refunded : 31200.0
Available Balance : 0.0
PS C:\java\oop exp-5>
```

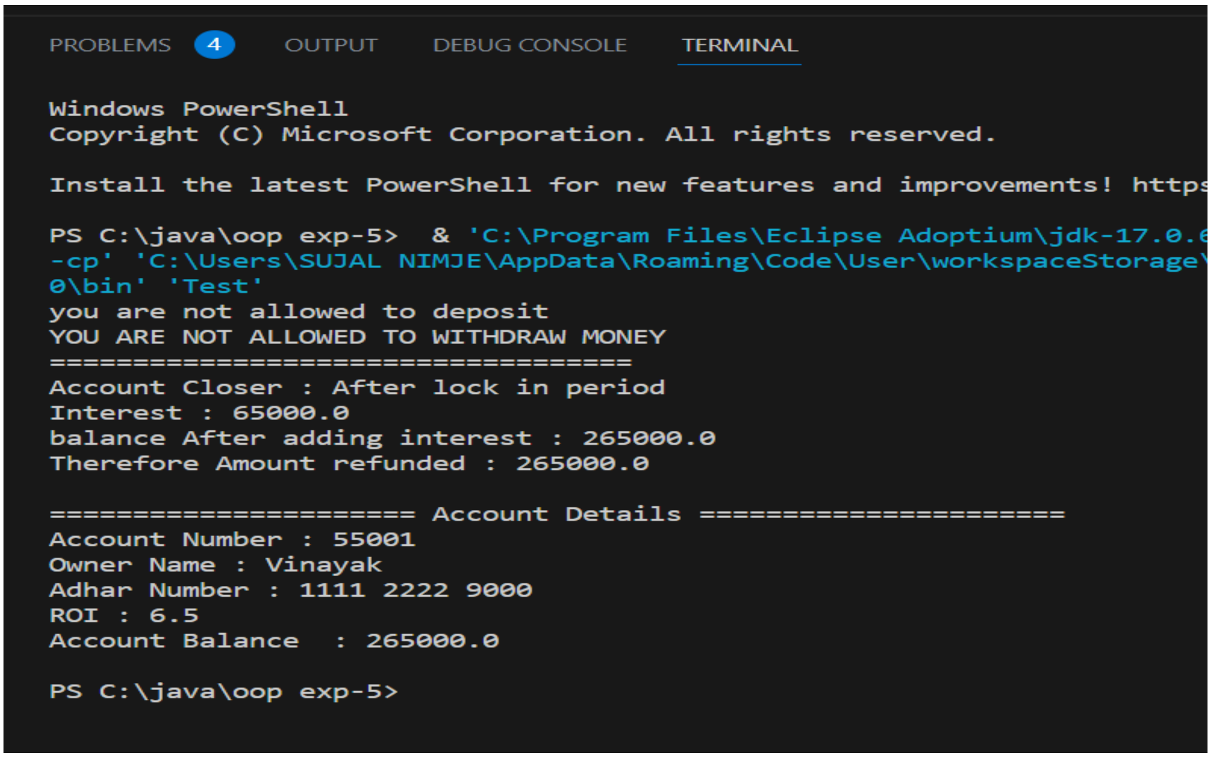
Date : 24-05-23

3. Create a FixedDeposit Account with an amount of 200000 and LockIn Period = 5 years Deposit Rs. 50000 to the account Withdraw 20000 from the account Display account details Close the account after 2 years of opening date



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! htt

PS C:\java\oop exp-5>  & 'C:\Program Files\Eclipse Adoptium\jdk-17.0
-cp' 'C:\Users\SUJAL NIMJE\AppData\Roaming\Code\User\workspaceStorag
0\bin' 'Test'
you are not allowed to deposit
YOU ARE NOT ALLOWED TO WITHDRAW MONEY

====================== Account Details ======================
Account Number : 55000
Owner Name : Bubli
Adhar Number : 1111 5678 9000
ROI : 0.0
Account Balance  : 200000.0

=====================================
Account Closer : before lock in period!
Penalty : 10000.0
Amount refunded : 190000.0
PS C:\java\oop exp-5>
```

4. *Same as above with Closing the account on completing 5 years after opening date Display account details*



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https

PS C:\java\oop exp-5>  & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.
-cp' 'C:\Users\SUJAL NIMJE\AppData\Roaming\Code\User\workspaceStorage
0\bin' 'Test'
you are not allowed to deposit
YOU ARE NOT ALLOWED TO WITHDRAW MONEY
===================================
Account Closer : After lock in period
Interest : 65000.0
balance After adding interest : 265000.0
Therefore Amount refunded : 265000.0

====================== Account Details ======================
Account Number : 55001
Owner Name : Vinayak
Adhar Number : 1111 2222 9000
ROI : 6.5
Account Balance  : 265000.0

PS C:\java\oop exp-5>
```