CAP:680 JAVA PROGRAMMING LANGUAGE LABORATORY

ASSIGNMENT-9

ST_NAME: - EKHLAKH AHMAD
REG NO.: - 12209166
ROLL NO.: - RD2215B50
SECTION: - D2215
GROUP: - 2

Q.1. All the banks operating in India are controlled by RBI. (e.g. minimum interest rate, minimum balance allowed, maximum withdrawal limit etc) which all banks must follow. Suppose RBI has set minimum interest rate applicable to a saving bank account to be 4% annually. However, banks are free to use 4% interest rate or to set any rates above it. Construct a program to implement bank functionality in the above scenario and demonstrate the dynamic polymorphism concept. Note: Create few classes namely Customer, Account, RBI (Base Class) and few derived classes (SBI, ICICI, PNB etc). Assume and implement required member variables and functions in each class.

Ans: -

```
abstract class RBI {
   protected double minInterestRate;

   public RBI(double minInterestRate) {
      this.minInterestRate = minInterestRate;
   }

   public abstract double calculateInterest(double balance);
}

class SBI extends RBI {
   private double interestRate;

   public SBI(double interestRate) {
      super(4.0);
```

```
        this.interestRate = interestRate;
    }

    @Override
    public double calculateInterest(double balance) {
        double rate = Math.max(minInterestRate, interestRate);
        return balance * rate / 100;
    }
}

class ICICI extends RBI {
    private double interestRate;

    public ICICI(double interestRate) {
        super(4.0);
        this.interestRate = interestRate;
    }

    @Override
    public double calculateInterest(double balance) {
        double rate = Math.max(minInterestRate, interestRate);
        return balance * rate / 100;
    }
}

class PNB extends RBI {
    private double interestRate;
```

```
    public PNB(double interestRate) {
        super(4.0);
        this.interestRate = interestRate;
    }

    @Override
    public double calculateInterest(double balance) {
        double rate = Math.max(minInterestRate, interestRate);
        return balance * rate / 100;
    }
}

class Account {
    private RBI bank;
    private double balance;

    public Account(RBI bank, double balance) {
        this.bank = bank;
        this.balance = balance;
    }

    public double getBalance() {
        return balance;
    }

    public void deposit(double amount) {
        balance += amount;
    }
```

```
    public boolean withdraw(double amount) {
        if (balance - amount < 0) {
            return false;
        }

        balance -= amount;
        return true;
    }

    public double calculateInterest() {
        return bank.calculateInterest(balance);
    }
}

class Customer {
    private String name;
    private Account account;

    public Customer(String name, Account account) {
        this.name = name;
        this.account = account;
    }

    public String getName() {
        return name;
    }
```

```
    public Account getAccount() {
        return account;
    }

    public void deposit(double amount) {
        account.deposit(amount);
    }

    public boolean withdraw(double amount) {
        return account.withdraw(amount);
    }

    public double calculateInterest() {
        return account.calculateInterest();
    }
}

public class AccountDemo {
    public static void main(String[] args) {
        SBI sbi = new SBI(4.5);
        ICICI icici = new ICICI(4.25);
        PNB pnb = new PNB(4.0);

        Account account1 = new Account(sbi, 10000);
        Account account2 = new Account(icici, 5000);
        Account account3 = new Account(pnb, 2000);

        Customer customer1 = new Customer("ekhlakh", account1);
```

```
        Customer customer2 = new Customer("sonpal", account2);
        Customer customer3 = new Customer("hamid", account3);

        System.out.println("SBI bank customer details:");
        System.out.println("Name: " + customer1.getName());
        System.out.println("Account balance: " +
customer1.getAccount().getBalance());
        System.out.println("Interest rate: " + sbi.calculateInterest(0));

        System.out.println("\nICICI bank customer details:");
        System.out.println("Name: " + customer2.getName());
        System.out.println("Account balance: " +
customer2.getAccount().getBalance());
          System.out.println("Interest rate: " + icici.calculateInterest(0));

    System.out.println("\nPNB bank customer details:");
    System.out.println("Name: " + customer3.getName());
    System.out.println("Account balance: " +
customer3.getAccount().getBalance());
    System.out.println("Interest rate: " + pnb.calculateInterest(0));

    // deposit some amount into customer1's account
    customer1.deposit(5000);

    // withdraw some amount from customer2's account
    boolean withdrawSuccess = customer2.withdraw(3000);
    if (withdrawSuccess) {
        System.out.println("\nWithdrawal successful.");
    } else {
```

```
        System.out.println("\nInsufficient balance for withdrawal.");
    }

    // calculate interest for all customers
    System.out.println("\nInterest for " + customer1.getName() + ": " +
customer1.calculateInterest());
    System.out.println("Interest for " + customer2.getName() + ": " +
customer2.calculateInterest());
    System.out.println("Interest for " + customer3.getName() + ": " +
customer3.calculateInterest());
}
}
```

OUTPUT

```
PROBLEMS 27   OUTPUT   DEBUG CONSOLE   TERMINAL

SBI bank customer details:
Name: ekhlakh
Account balance: 10000.0
Interest rate: 0.0

ICICI bank customer details:
Name: sonpal
Account balance: 5000.0
Interest rate: 0.0

PNB bank customer details:
Name: hamid
Account balance: 2000.0
Interest rate: 0.0

Withdrawal successful.

Interest for ekhlakh: 675.0
Interest for sonpal: 85.0
Interest for hamid: 80.0
PS D:\VS CODE\JAVA PROGRAM\ASSIGNMENT>
```