DATE / /
PAGE

LAB PROGRAMME-5

Develop a java programme to create a class Bank that maintains two kinds of account for it's customers one called savings account and other current account. The saving account provides compound Interest and withdrawl but not cheque book Facility, the current account provides cheque book facility but no interest Current account holders also maintain Minimum balance if balance falls below this level a Service is imposed

```
import java.util.Scanner;

class Account {
  protected String custName;
  protected int accNo;
  protected String accType;
  protected double balance;

  public Account (String customerName, int accNo,
  String accountType, double balance) {
    this.customerName = customerName;
    this.accountNumber = accountNumber;
    this.accountType = accountType;
    this.balance = balance;
}
```

```
Public void deposit (double amount)
{ if (amount >0 ) {
    balance += amount;
    System.out.println("Deposit success, newbal:
  }
  else {
    system.out.println("Invalid deposit amt
  }
}

Public void displayBalance() {
  System.out.println("Acc balance: ₹"+balance);
  }
}

class CurrAcct extend Account {
  private static final double MinBal =500.0
  private static final double SERVICE CHA = 50;

  public ~~stat~~ CurrAcct (String customerName, int
  accountNumber, double balance) {
    super (customerName, account Number, "current",
    balance);
  }
```

```
public void withdraw (double amount) {
    if (amount >0 && amount <= balance) {
        balance -= amount;
        System.out.println ("withdrawal successful");
        checkMinimumBalance();
    }
    else {
        System.out.println ("Invalid withdrawal");
    }
}

private void checkMinimumBalance () {
    if (balance < MIN_BALANCE) {
        balance -= SERVICECHARGE;
        System.out.println ("Balance fell below minimum.
Service_CHARGE of ₹" + SERVICE_CHARGE + " applied");
    }
  }
}


class SavAcct extends Account {
    private static final double INTEREST_RATE=0.04;
    public SavAcct (String customerName, int
    accountNumber, double balance) {
        super (customerName, accountNumber, "Savings",
        balance);
    }

public void computeANDDepositInterest ()
{ double interest = balance * INTEREST_RATE;
```

```
        balance += interest;
        System.out.println("Interest of ₹" +
        interest + "has been added");
    }

    public void withdraw(double amount){
        if(amount > 0 && amount <= balance)
        { balance -= amount;
          System.out.println("withdrawal success");
        }
        else
        { system.out.println("Invalid withdraw");
        }

    }

}

class
public class Bank{
    public static void main(String[] args)
    { Scanner scanner = new Scanner(System.in);
      Account account = null;
      System.out.println("welcome to Bank");
      System.out.println("please select type");
      System.out.println("1.saving \n2.current");
```

```
int accountChoice = scanner.nextInt();
scanner.nextLine();

System.out.print("Enter your name: ");
String customerName = scanner.nextLine();
System.out.print("Enter your account Number: ");
int accountNumber = scanner.nextInt();
System.out.print("Enter initial amount");
double balance = scanner.nextDouble();

switch (accountChoice){
  case 1:
    account = new SavAcct (customerName, accountNumber,
    balance);
    System.out.println("Saving account created");
    break;
  case 2:
    account = new CurAcct (customerName, accountNumber,
    balance);
    System.out.println("current Account created");
    break;
  default:
    System.out.println("Invalid choice.");
    return;

}
```

```
while (true) {
    System.out.println("\n choose operation");
    Sout ("1. Deposit")
    System.out.println("2. Display Balance \n3.
    Compute and Deposit Interest (Saving Acct) \n
    4. withdraw \n 5. exit");

    int operationChoice = scanner.nextInt();

    switch (operationchoice) {
    case 1:
        System.out.print("Enter deposit");
        double depositAmount = scanner.nextDouble();
        account.deposit(depositAmount);
    case 2:
        account.displayBalance();
        break;
    case 3:
        if (account instance of SavAcct)
        { (SavAcct) account).computeAndDepositInter
        }
        else
        { System.out.print("calculation is
        only for Saving account");
        }
        break;
```

DATE / /
PAGE

```
case 4:
    System.out.println("Enter withdraw Amount");
    double withdrawlAmount = scanner.nextDouble();
    if (account instanceof SavAcct) {
        ((SavAcct) account).withdraw(withdrawl
Amount);
    }
    else if (account instanceof currAct) {
        ((CurrAcct) account).withdraw(withdrawlAmount);
    }
    break;

case 5:
    System.out.println("Thanks for banking");
    return;
default:
    System.out.println("Invalid choice");
    }
  }
}
```

DATE / /
PAGE

Output

please select type of account you want to create:
1. Savings Account
2. Current Account
1
Enter your name: soham
Enter your account number: 1122
Enter initial deposit amount: 112
saving Account created successfully!

Choose an operation:
1. Deposit
2. Display balance
3. Compute and Deposit Interest (Saving Acct)
4. withdrawal
5. exit
1
Enter deposit amount: 123
Deposit successful new balance 235.0

Choose an operation:
1. Deposit
2. Display balance
3. compute and deposit Interest (Saving Acct)
4. withdrawal
5. exit
5
Thanks for Banking with us

07/11/24

Lab programme 5

```
import java.util.Scanner;

class Account {
    protected String customerName;
    protected int accountNumber;
    protected String accountType;
    protected double balance;

    public Account(String customerName, int accountNumber,
    String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. New balance: " + balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

    public void displayBalance() {
        System.out.println("Account Balance: " + balance);
    }
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.07;

    public SavAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Savings", balance);
    }

    public void computeAndDepositInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest of " + interest +
        " has been added. New balance: " + balance);
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
```

```
            System.out.println("Withdrawal successful. New balance: $" + balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    }
}

class CurAcct extends Account {
    private static final double MIN_BALANCE = 500.0;
    private static final double SERVICE_CHARGE = 50.0;

    public CurAcct(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Current", balance);
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
            checkMinimumBalance();
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    }

    private void checkMinimumBalance() {
        if (balance < MIN_BALANCE) {
            balance -= SERVICE_CHARGE;
            System.out.println("Balance fell below minimum. Service charge of "
            + SERVICE_CHARGE + " applied. New balance: " + balance);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter customer name: ");
        String customerName = scanner.nextLine();

        System.out.print("Enter account number: ");
        int accountNumber = scanner.nextInt();

        System.out.print("Enter initial balance: ");
        double initialBalance = scanner.nextDouble();

        System.out.println("Choose account type:");
        System.out.println("1. Savings Account");
        System.out.println("2. Current Account");
        int accountChoice = scanner.nextInt();
```

```
Account account;
if (accountChoice == 1) {
    account = new SavAcct(customerName, accountNumber, initialBalance);
} else if (accountChoice == 2) {
    account = new CurAcct(customerName, accountNumber, initialBalance);
} else {
    System.out.println("Invalid account type selection.");
    scanner.close();
    return;
}

boolean exit = false;
while (!exit) {
    System.out.println("\nChoose an operation:");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    if (account instanceof SavAcct) {
        System.out.println("4. Compute and Deposit Interest ");
    }
    System.out.println("5. Exit");
    int choice = scanner.nextInt();

    switch (choice) {
        case 1:
            System.out.print("Enter amount to deposit: ");
            double depositAmount = scanner.nextDouble();
            account.deposit(depositAmount);
            break;

        case 2:
            System.out.print("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            if (account instanceof SavAcct) {
                ((SavAcct) account).withdraw(withdrawAmount);
            } else if (account instanceof CurAcct) {
                ((CurAcct) account).withdraw(withdrawAmount);
            }
            break;

        case 3:
            account.displayBalance();
            break;

        case 4:
            if (account instanceof SavAcct) {
                ((SavAcct) account).computeAndDepositInterest();
            } else {
                System.out.println("Invalid choice for Current Account.");
            }
```

```
                    break;

                case 5:
                    exit = true;
                    System.out.println("Exiting...");
                    break;

                default:
                    System.out.println("Invalid choice.");
            }
        }

        scanner.close();
    }
}
```

OUTPUT

Enter customer name:xyz
Enter account number: 1001
Enter initial balance: 1000
Choose account type:

1. Savings Account
2. Current Account
   1

Choose an operation:

1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
   1
   Enter amount to deposit: 200
   Deposit successful. New balance: 1200.0

Choose an operation:

1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
   4
   Interest of 48.0 has been added. New balance: 1248.0

Choose an operation:

1. Deposit
2. Withdraw

3. Display Balance
4. Compute and Deposit Interest
5. Exit
   5
   Exiting...