09/01/24 LAB- "05" :-

Q. Create a class Account that stores: customer name, account number and type of account. From this derive the classes Current-account and Saving-account to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

(a) Accept deposit from customer and update the balance.
(b) Display the balance
(c) Compute and deposit interest
(d) Permit withdrawal and update the balance

- Check for the minimum balance, impose penalty if necessary and update the balance

Sol?

```
import java.util.Scanner;

Class Amount
{
  String customerName;
  int accountNumber;
  String accountType;
  double balance;

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

```
    this.balance = balance;
  }
void deposit (double amount)
  {
   balance += amount;
   System.out.println ("Deposit of" + amount
 successful");
  }

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

 void withdraw (double amount)
  {
    if (balance - amount < 0)
    {
     System.out.println ("Insufficient balance");
     return;
    }
    balance -= amount;
    System.out.println ("Withdrawal of "
     + amount + " Successful");
    }
}

class SavingAccount extends Account
  {
   SavingAccount (String customerName,
int accountNumber, string accountType,
double balance)
```

```
{
super (customerName, accountNumber, accountType,
  balance);
}

Void compoundInterest()
 {
  double rate = 0.05;
  double time = 1.0;
  double interest = balance * Math.pow(1+
  rate, time) - balance;
  balance += interest;
  System.out.println("Interest of " + interest +
    " added");
 }

Void withdraw (double amount)
 {
  if (balance - amount < 0)
  {
    System.out.println("Insufficient
                        balance.");
    return;
  }
  balance -= amount;
  System.out.println("withdrawal of "
  + amount + " successful");
  }
}

class CurrentAccount extends Account
  {
```

```
    double minimumBalance = 1000;
    double ServiceCharge = 50;

    CurrentAccount (String CustomerName,
int accountNumber, String accountType,
double balance)
    {
    Super (CustomerName, accountNumber,
    accountType, balance);
    }

  Void withdraw (double amount)
    {
    if (balance - amount < minimumBalance)
    {
      System.out.println ("Insufficient
                          balance");
      return;
    }
    balance -= amount;
  System.out.println ("Withdrawal of "
  + amount + " Successful");
    }

  Void imposeServiceCharge ()
    {
      if (balance < minimumBalance)
      {
        balance -= ServiceCharges;
        System.out.println ("Service
  Charge of " + ServiceCharge + " imposed");
    }
  }
}
```

```
public class Bank
{
    public static void main (String args [])
    {
        Scanner scanner = new Scanner (System.in);
        System.out.println ("Enter Customer name: ");
        String CustomerName = Scanner.nextLine ();
        System.out.println ("Enter account number: ");
        int accountNumber = scanner.nextInt();
        System.out.println ("Enter account type
            (Savings / Current): ");
        String accountType = scanner.next();
        System.out.println ("Enter initial balance: ");
        double balance = scanner.nextDouble();

        Account account;
        if (AccountType.equals ("Savings"))
        {
            account = new SavingsAccount
            (CustomerName, accountNumber,
            accountType, balance);
        }
        else
        {
            account = new CurrentAccount
            (CustomerName, accountNumber,
            accountType, balance);
        }

while (true)
```

```
{
System.out.println("\n 1. Deposit");
System.out.println("\n 2. Display balance");
System.out.println("3. Compute and
                    deposit interest");
System.out.println("4. Withdrawal");
System.out.println("5. Exit");
System.out.println("Enter choice:");
int choice = Scanner.nextInt();

Switch (choice)
 {
 Case 1:
  System.out.println("Enter amount to
   be deposit");
  double amount = scanner.nextDouble
   ();
  account.deposit (amount);
  break;
 Case 2:
  account.displayBalance();
  break;
 Case 3:
  if (account instance of SavingAccount)
  {
  ((SavingAccount) account).Compound
   Interest();
  }
  else
  {
  System.out.println("Interest not
available for current account");
  }
```

```
break;

Case 4:
 System.out.println("Enter amount to
 withdraw: ");
 amount = Scanner.nextDouble();
 account.withdraw(amount);
 if (account instance of CurrentAccount)
 {
 ((CurrentAccount) account).imposeService
 Charge();
 }
 break;

Case 5:
 System.exit(0);
 }
 }
 }
} //.
```

:- Output :-

```
Enter customer name: Prajwal K
Enter account number: 199
Enter account type (savings/Current): savings
Enter initial balance: 2840

1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdraw
5. Exit
```

Enter choice: 3
Interest of 144.5 added

1. Deposit
2. Display balance
3. Compute and deposit interest
4. Withdraw
5. Exit
Enter choice: 4
Enter amount to withdraw: 1770
withdraw of 1770.0 successful

1. Deposit
2. Display balance
3. Compute and deposit interest
4. withdraw
5. Exit
Enter choice: 2
Balance: 1264.5

1. Deposit
2. Display balance
3. Compute and deposit interest
4. withdraw
5. Exit
Enter choice: 5

Thu
11/01/24