PAGE NO :
DATE :

Week-5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

```
import java.util.Scanner;

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

Account(String name, int accNumber, String accType)
{   customerName = name;
    AccountNumber = accNumber;
    accountType = accType;
    balance = 0.0;
}

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

```
    void displayBalance() {
        System.out.println("Account balance for " +
            customerName + ": Rs" + balance);
    }
}

class CurrAcct extends Account {
    double minimumBalance;
    double serviceCharge;

    CurrAcct(String name, int accNumber,
        double minBalance) {

        super(name, accNumber, "Current");

        minimumBalance = minBalance;
        serviceCharge = 10.0;
    }

    void checkMinimumBalance() {

        if(balance < minimumBalance) {
            balance += serviceCharge;

            System.out.println("Service charge of Rs" +
            serviceCharge + " applied. New Balance: Rs" +
            balance);
        }
    }
```

```
    void displayBalance() {
        System.out.println("Account balance for " +
            customerName + ": Rs " + balance);
    }
}

class CurrAcct extends Account {
    double minimumBalance;
    double serviceCharge;

    CurrAcct(String name, int accNumber,
        double minBalance) {

    super(name, accNumber, "Current");

    minimumBalance = minBalance;
    serviceCharge = 10.0;
}

void checkMinimumBalance() {

    if(balance < minimumBalance) {
        balance += serviceCharge;

        System.out.println("Service charge of Rs" +
        serviceCharge + "applied. New Balance: Rs" +
        balance);
    }
}
```

```
void withdraw(double amount) {
  if(amount <= balance) {
      balance -= amount;
    System.out.println("Withdrawal of Rs" +
    amount + " successful. New Balance: Rs" +
    balance);
}
else { System.out.println("Insufficient
  funds. Withdrawal failed.");
}
}
}

class SavAcct extends Account {
    double interestRate;

  SavAcct(String name, int accNumber, double
            interestRate) {

super(name, accNumber, "Savings");

this.interestRate = interestRate;
}
void computeInterest() {
  double interest = balance * (interestRate/100);
  balance += interest;

System.out.println("Interest of Rs" + interest +
"applied. New Balance: Rs" + balance); }
else { System.out.println("Insufficient funds"
+ "Withdrawal failed");
}}}
```

```
public class Bank {
  public static void main (String[] args)
  {
    Scanner scanner = new Scanner (System.in);
    System.out.println("Enter your name :");

    String customerName = scanner.nextLine();

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

    CurrAcct currentAccount = new CurrAcct(customerName,
    accountNumber, 1000.0);

    SavAcct savingsAccount = new SavAcct(customerName,
    accountNumber, 5.0);

    int choice;
    do { System.out.println("\nMenu:");
         System.out.println("1. Deposit");

         System.out.println("2. Withdraw");
         System.out.println("3. Compute interest");
         System.out.println("4. Display Account details");
         System.out.println("5. Exit");
         System.out.print("Enter your choice:");
         choice = scanner.nextInt();

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

```
case 2: System.out.printC("Enter withdrawal
            amount:");
double withdrawAmount = scanner.nextDouble();
currentAccount.withdraw(withdrawAmount);
break;

case 3: savingsAccount.computeInterest();
            break;

case 4: currentAccount.displayBalance(); break;

case 5: System.out.println("Exiting progeram.
            Thank you!"); break;

default: system.out.println("Invalid choice. Please
            enter a valid option.");
    }
} while (choice != 5);
}
}
```

Output: Enter customer name : Rahul
Enter account Number : 78
Enter customer name : Roshan
Enter account Number : 18

--MENU--
1. Deposit
2. Withdraw
3. Compute interest for savings Account
4. Display account details
5. Exit

Enter your choice : 1
Enter the type of account : saving
Enter the deposit amount : 1000

--Menu--
1. Deposit
2. Withdraw
3. Compute interest for Savings
4. Display Account details
5. Exit

Enter your choice = 2
Enter the withdrawal amount : 200

--Menu--
1. Deposit
2. withdraw
3. Compute interest for Savings Account
4. Display Account details
5. Exit

Enter your choice = 4
Customer name = Rahul
Account number = 78
Type of account = Saving
Balance = 800.0

10/01/24