SURYA Gold
Date Page

dt =

Lab-5
Bank program

```
import java.util.Scanner;
class Account{
    String customerName;
    int accountNumber;
    String accountType;
    double balance;

    Account(String customerName, int accountNumber,
            String accountType, double balance){
        this.customerName = customerName;
        this.accountName = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }
    void deposit(int amount){
        this.balance += amount;
        Sout("deposit =" + amount + "balance =" + this.balance);
    }
    void displayBalance(){
        Sout("acc balance =" + balance);
    }
}
```

```
class CurAccount extends Account{
    int minBalance;
    int ServiceCharge;
    CurAccount(String customerName,
               int accountNumber, int balance,
               int minBalance, int ServiceCharge){
        super(customerName, accountNumber,
              "current", balance);
        this.minBalance = minBalance;
        this.ServiceCharge = ServiceCharge;
    }
    void checkMinBalance(){
        if(balance < minBalance){
            balance -= ServiceCharge;
            Sout("min not maintains
                 .Service charge" + ServiceCharge
                 + "Imposed.");
            displayBalance();
        }
    }
    void withdraw(int amount){
        if(amount > balance){
            Sout("Insufficient funds");
        }else{
            balance -= amount;
            Sout("withdrawal of" + amount +
                 "balance" + balance);
            checkMinBalance();
        }
    }
}
```

```
class SavAccount extends Account {
    double interestRate;

    public SavAccount(String customerName,
                      double accountNumber,
                      double balance,
                      double interestRate)
    {
        super(customerName, accountNumber,
              "Savings", balance);
        this.interestRate = interestRate;
    }

    void computeInterest() {
        double interest = balance * interestRate/100;
        balance += interest;
        Sout("Interest = " + interest);
        displayBalance();
    }

    void withdraw(double amount) {
        if (amount > balance) {
            sout("Insufficient funds.");
        } else {
            balance -= amount;
            System.out.println("withdrawn = " + amount
                + " Successful. balance = "
                + balance);
        }
    }
}
```

```
Public class Bank {
    PSVM() {
        Scanner in = new Scanner(System.in);

        CurAccount CA = new CurAccount
            ("nishanth", 123, 1000
             , 500, 10);
        SavAccount SA = new SavAccount
            ("nishanth", 321, 2000, 5);
        int choice;
        do {
            Sout("Select an option:
                \n 1. deposit
                \n 2. display Balance
                \n 3. Compute Interest
                \n 4. withdraw:
                \n 5. Exit);
            Sout("enter choice:");
            choice = in.nextInt();
            switch (choice) {
                case 1:
                    Sout("enter amount");
                    int deposit = in.nextInt();
                    System.out.println("1. current 2.saving")
                    int acctype = in.nextInt();
                    if (acctype == 1) {
                        CurrentAccount.deposit(deposit);
                    } else if (acctype == 2) {
                        SavingsAccount.deposit(deposit);
                    } else {
                        Sout("Invalid");
                    }
                    break;
```

```
case 2:
    Sout ("Select acc (1, 2): ");
    int acc_type = in.nextInt();
    if (acc_type == 1) {
        CA.displayBalance();
    } else if (acc_type == 2) {
        SA.displayBalance();
    } else {
        Sout ("Invalid acc");
    }
    break;
case 3:
    if (SavingsAccount instance of SavAccount) {
        ((SavAccount) SA).computeInterest();
    } else {
        Sout ("Invalid option for
               current account")
    }
    break;

case 4:
    Sout ("enter withdrawal");
    int WA = in.nextInt();
    Sout ("1. current, 2. Savings");
    int acc = in.nextInt();
    if (acc == 1) {
        CA.withdraw(WA);
    } else if (acc == 2) {
        SA.withdraw(WA);
    } else {
        Sout ("Invalid")
    break;
```

```
case 5:
    Sout ("Thanks");
    break;
default:
    Sout (" Invalid. ");
}
} while (choice != 5);
scanner.close();
}
}
```

O/P

```
1. Deposit
2. balance
3. Interest
4. withdraw
5. exit
choice: 1
deposit = 50000
1. current or 2. Saving
Deposit of 50000, balance = 501000.0

2
<current 2. Saving>
501000.0
```

write all cases
Result
9/1/2024