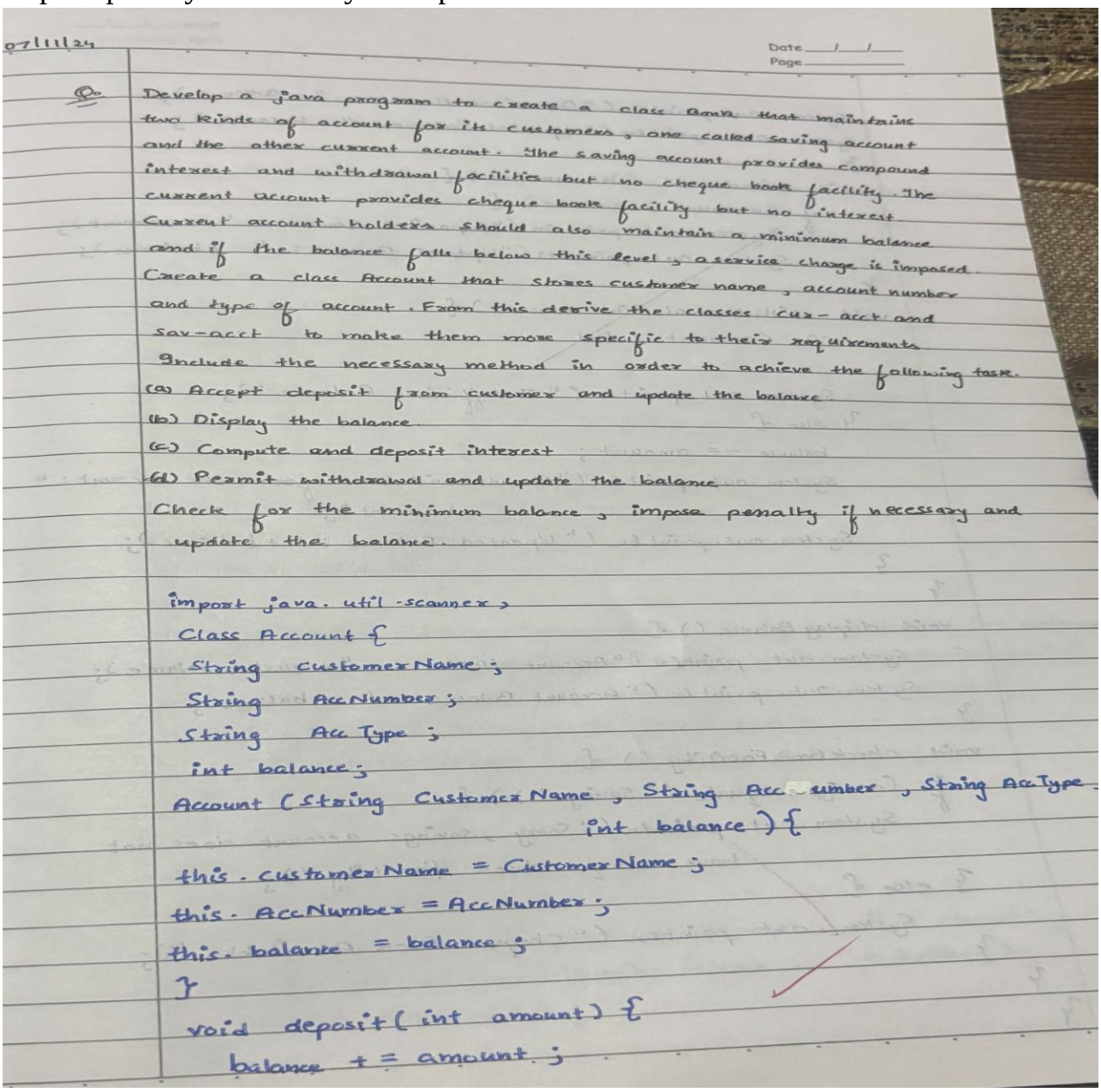
LAB 05

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. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct 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.



07/11/24

Date __/__/__
Page

Q. Develop a java program to create a class Bank that maintains two kinds of account for its customers, one called saving account and the other current account. The saving 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. Create a class Account that stores customer name, account number and type of account. From this derive the classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary method in order to achieve the following task.

(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.

```
import java.util.Scanner;
Class Account {
String customerName;
String AccNumber;
String AccType;
int balance;
Account (String CustomerName, String AccNumber, String AccType,
                                    int balance) {
this.customerName = CustomerName;
this.AccNumber = AccNumber;
this.balance = balance;
}
void deposit(int amount) {
  balance += amount;
```

```
System.out.println("Deposited Amount: " + amount);
System.out.println("Updated Balance: " + balance);
}

void withdrawal(int amount) {
    if (amount > balance) {
        System.out.println("Deposited Amount: " + amount);
        System.out.println("Updated Balance: " + balance);
}

void withdrawal(int amount) {
    if (amount > bal

        System.out.println("Insufficient Balance");
    } else {
        balance -= amount;
        System.out.println("Amount withdrawn from the account: "
                           + amount);
        System.out.println("Updated Balance: " + balance);
    }
}

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

void checkBookFacility() {
    if (accType.equals("Savings")) {
        System.out.println("Sorry, savings account does no
                           have cheque book facility");
    } else {
        System.out.println("Cheque Book Available.");
    }
}
}
```

```
Class SavingAccount extends Account {
    int interestRate;

SavingAccount (String customerName, String accNumber, int balance,
                int interestRate) {
    super (customerName, accNumber, "Savings", balance);
    this.interestRate = interestRate;
}
void computeAndDepositInterest () {
    int interest = balance * interestRate/100;
    balance += interest;
    System.out.println ("Interest Earned: " + interest);
    System.out.println ("Updated Balance: " + balance);
}
}
Class CurrentAccount extends Account {
    int minimumBalance;
    int serviceCharge;

CurrentAccount (String customerName, String accNumber, int balance,
                int minimumBalance, int serviceCharge) {
    super (customerName, accNumber, "Current", balance);
    this.minimumBalance = minimumBalance;
    this.serviceCharge = serviceCharge;
}

void deposite (int amount) {
    balance += amount;
    System.out.println ("Deposited Amount: " + amount);
    System.out.println ("Updated Balance: " + balance);
    CheckForMinimum ();
}
```

```
void withdrawal (int amount) {
    if (amount > balance) {
        System.out.println ("Insufficient Balance");
    } else {
        balance -= amount;
        System.out.println ("Amount withdrawn from the
                            account : " + amount);
        System.out.println ("Updated Balance: " +balance);
        CheckForMinimum ();
    }
}

void checkForMinimum () {
    if (balance < minimumBalance) {
        balance -= serviceCharge;
        System.out.println ("Account Balance after imposing
                            service charge : " + balance);
    }
}

Class Bank {
    public static void main (String [] args) {
SavingAccount s1 = new SavingAccount ("Milan", "123", 5000, 13);
System.out.println ("For Saving Account :");
s1.displayBalance ();
s1.computeAndDepositInterest ();
s1.deposit (1000);
s1.withdrawal (2000);
s1.displayBalance ();

CurrentAccount c1 = new CurrentAccount ("Rahul", "456", 10
                    2500, 100);
System.out.println ("For Current Account :");
```

```
c1.displayBalance();
c1.deposit(1000);
c1.withdrawal(500);
c1.displayBalance();
}
}
```

Output

For Saving Account:
Account Holder : Milan
Account Balance : 5000
Interest Earned : 650
Updated Balance : 5650
Deposited Amount : 1000
Updated Balance : 6650
Amount withdrawn from the account : 2000
Updated Balance : 4650
Sorry, Saving account does not have cheque book facility
Account Holder : Milan
Account Balance : 4650
For Current Account :
Account Holder : Rahul
Account Balance : 100
Deposited Amount : 1000
Updated Balance : 1100
Account Balance after imposing service charge : 1000
Account withdrawn from the account : 500
Updated Balance : 500
Account Balance after imposing service charge : 400
Account Holder : Rahul
Account Balance : 400

21/11/24

```
import java.util.Scanner;

abstract class Account {
    String customerName, accountNumber;
    double balance;

    Account(String customerName, String accountNumber, double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }

    abstract void deposit(double amount);
    abstract void displayBalance();
    abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

    SavAcct(String customerName, String accountNumber, double initialBalance, double
interestRate) {
        super(customerName, accountNumber, initialBalance);
        this.interestRate = interestRate;
    }

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

    void displayBalance() {
```

```
        System.out.println("Savings Balance: " + balance);
    }

    void withdraw(double amount) {
        if (amount <= balance) balance -= amount;
    }

    void computeAndDepositInterest() {
        balance += balance * interestRate / 100;
    }
}

class CurAcct extends Account {
    static final double MIN_BALANCE = 1000, SERVICE_CHARGE = 50;

    CurAcct(String customerName, String accountNumber, double initialBalance) {
        super(customerName, accountNumber, initialBalance);
    }

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

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

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
```

```
            if (balance < MIN_BALANCE) balance -= SERVICE_CHARGE;
        }
    }
}

class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter account type (savings/current): ");
        String type = scanner.nextLine();

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

        System.out.println("Enter account number: ");
        String number = scanner.nextLine();

        Account account;
        if (type.equals("savings")) {
            System.out.println("Initial balance and interest rate: ");
            account = new SavAcct(name, number, scanner.nextDouble(), scanner.nextDouble());
        } else {
            System.out.println("Initial balance: ");
            account = new CurAcct(name, number, scanner.nextDouble());
        }

        while (true) {
            System.out.println("\n1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit");
            int choice = scanner.nextInt();
            switch (choice) {
```

```
            case 1: account.deposit(scanner.nextDouble());
            break;
            case 2: account.displayBalance();
            break;
            case 3: account.withdraw(scanner.nextDouble());
            break;
            case 4: if (account instanceof SavAcct) ((SavAcct)
account).computeAndDepositInterest();
            break;
            case 5:
            return;
        }
    }
  }
}
```

```
D:\24BMSCE>javac Bank.java

D:\24BMSCE>java Bank
Enter account type (savings/current):
savings
Enter customer name:
anu rai
Enter account number:
123786645087301
Initial balance and interest rate:
5000
50

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
1
200

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
2
Savings Balance: 5200.0

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
3
100

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
4

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
2
Savings Balance: 7650.0

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
5
```