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 facili es 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.

Source Code :

```
import java.util.Scanner;

class Account {
    String customerName;      int
accountNumber;      String
accountType;      double balance;
    public Account(String customerName, int accountNumber, String accountType) {
this.customerName = customerName;        this.accountNumber = accountNumber;        this.accountType =
accountType;        this.balance = 0.0;
    }
   public void deposit(double amount) {        if
(amount > 0) {            balance += amount;
          System.out.println("Amount deposited: " + amount);          System.out.println("Updated balance: " +
balance);
       } else {
          System.out.println("Invalid deposit amount!");
```

```
        }
    }
    public void displayBalance() {
        System.out.println("Balance: " + balance);
    }
}  class SavAcct extends Account {
private double interestRate;
    public SavAcct(String customerName, int accountNumber, double interestRate) {        super(customerName,
accountNumber, "Savings");         this.interestRate = interestRate;
    }      public void computeAndDepositInterest() {         double interest =
balance * (interestRate / 100);         balance += interest;
        System.out.println("Interest added: " + interest);         System.out.println("Updated balance: " + balance);
    }      public void withdraw(double amount) {         if
(amount <= balance) {             balance -= amount;
            System.out.println("Amount withdrawn: " + amount);             System.out.println("Updated balance: " +
balance);
        } else {
            System.out.println("Insufficient balance!");
        }
    }
}  class CurAcct extends Account {
double minimumBalance;     double
serviceCharge;
    public CurAcct(String customerName, int accountNumber, double minimumBalance, double serviceCharge)
{
        super(customerName,              accountNumber,              "Current");
this.minimumBalance = minimumBalance;              this.serviceCharge =
serviceCharge;
    }
```

```
   public void withdraw(double amount) {        if
(amount <= balance) {            balance -= amount;
          System.out.println("Amount withdrawn: " + amount);            if (balance <
minimumBalance) {                imposePenalty();
          }
          System.out.println("Updated balance: " + balance);
       } else {
          System.out.println("Insufficient balance!");
       }
   }
   private void imposePenalty() {        balance -=
serviceCharge;
       System.out.println("Balance fell below minimum. Service charge imposed: "
+ serviceCharge);
   }
}  public class Bank {
   public static void main(String[] args) {
       Scanner scanner = new Scanner(System.in);
       System.out.println("Choose account type:\n1. Savings Account\n2. Current Account");        int choice =
scanner.nextInt();        scanner.nextLine();

       System.out.println("Enter customer name: ");
       String name = scanner.nextLine();
       System.out.println("Enter account number: ");        int accNum =
scanner.nextInt();

       if (choice == 1) {
          System.out.println("Enter interest rate for savings account: ");            double interestRate =
scanner.nextDouble();
          SavAcct savAccount = new SavAcct(name, accNum, interestRate);
          System.out.println("Enter amount to deposit: ");            double deposit =
scanner.nextDouble();            savAccount.deposit(deposit);

          savAccount.computeAndDepositInterest();
          System.out.println("Enter amount to withdraw: ");
```

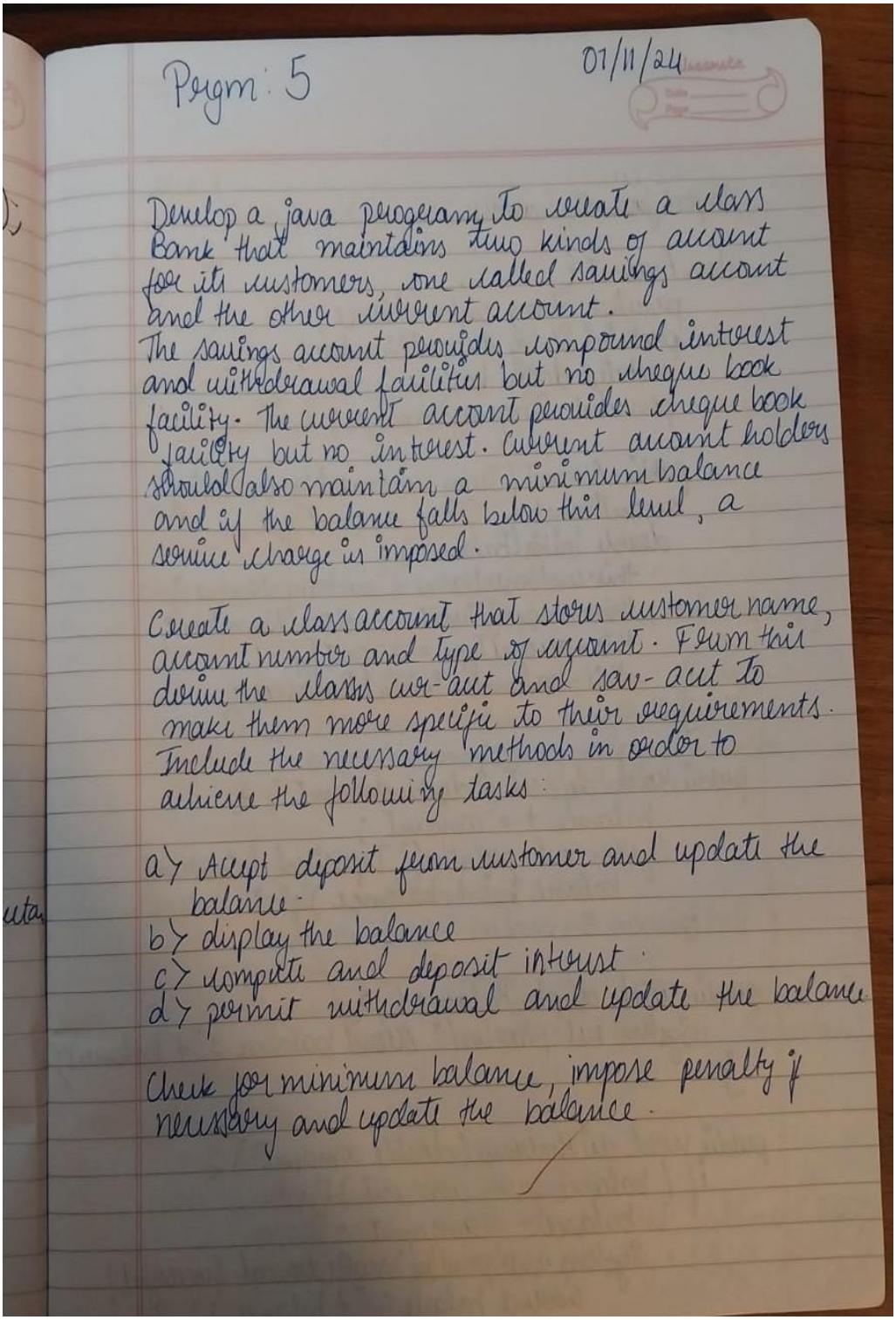
```
            double withdrawAmount = scanner.nextDouble();
            savAccount.withdraw(withdrawAmount);
        } else if (choice == 2) {
            System.out.println("Enter minimum balance for current account: ");            double minBalance =
scanner.nextDouble();
            System.out.println("Enter service charge for falling below minimum balance: ");
            double serviceCharge = scanner.nextDouble();
            CurAcct curAccount = new CurAcct(name, accNum, minBalance, serviceCharge);

            System.out.println("Enter amount to deposit: ");            double deposit =
scanner.nextDouble();            curAccount.deposit(deposit);

            System.out.println("Enter amount to withdraw: ");            double withdrawAmount =
scanner.nextDouble();            curAccount.withdraw(withdrawAmount);

        } else {
            System.out.println("Invalid account type selected.");
        }
        scanner.close();
    }
}
```

Output :

```
Choose account type:
1. Savings Account
2. Current Account
1
Enter customer name:
sagar
Enter account number:
1234
Enter interest rate for savings account:
3
Enter amount to deposit:
5000
Amount deposited: 5000.0
Updated balance: 5000.0
Interest added: 150.0
Updated balance: 5150.0
Enter amount to withdraw:
4800
Amount withdrawn: 4800.0
Updated balance: 350.0
```

```
Choose account type:
1. Savings Account
2. Current Account
2
Enter customer name:
chetan
Enter account number:
9876
Enter minimum balance for current account:
1000
Enter service charge for falling below minimum balance:
150
Enter amount to deposit:
6000
Amount deposited: 6000.0
Updated balance: 6000.0
Enter amount to withdraw:
5200
Amount withdrawn: 5200.0
Balance fell below minimum. Service charge imposed: 150.0
Updated balance: 650.0
```

OBSERVATION:



Pgm: 5 01/11/24

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 minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;

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

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

public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited! Current
            balance: " + balance);
    }

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

public void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("withdrawal Successful!
            current balance: " + balance); }
```

```
        else {
            System.out.println("Insufficient balance!");
        }
    }

    public String getAccountType() {
        return accountType;
    }
}

class SavAcct extends Account {
    private static final double interestRate = 0.04;

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

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

class curAcct extends Account {
    private static final double min_balance = 500;
    private static final double penalty = 50;
```

```
publi curAct (String customerNam, String accountNumb
    double initialBalance) {
        super(customerName, accountNumber,
            "current", initialBalance);
    }
publi void checkminimumBalance(){
    if(balance < min_balance) {
            balance -= penalty;
            System.out.println("Balance is below
                minimum. penalty added/deduct
            new balance: " + balance);
        }
    }
}

publi class Bank1 {
    publi 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 type (1 for
            Savings, 2 for Current): ");
        int accountChoice = scanner.nextInt();

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

        Account account = null;
```

```
if (accountChoice == 1) {
    System.out.print("Enter initial deposit for
        Savings account : ");
    double initialDeposit = scanner.nextDouble();
    account = new SavAcct(customerName,
    accountNumber, initialDeposit);
} else if (accountChoice == 2) {
    System.out.print("Enter initial deposit for
    current account : ");
    double initialDeposit = scanner.nextDou
    account = new CurAcct(customerName, accountNumb
        initialDeposit);
} else {
    System.out.println("Invalid");
    return;
}

boolean running = true;
while (running) {
    System.out.println("\n Bank Operations : ");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    System.out.println("4. Compute Interest");
    System.out.println("5. Check and apply
        minimum balance penalty");
    System.out.println("6. Exit");
    System.out.print("Enter your choice : ");
    int choice = scanner.nextInt();

    switch (choice) {
```

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

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

case 3:
    account.displayBalance();
    break;

case 4:
    if (account instanceof SavAct) {
        ((SavAct)account).computeInterest();
    } else {
        System.out.println("Interest can be
calculated only for savings account.");
    }
    break;

case 5:
    if (account instanceof CurAct) {
        ((CurAct)account).checkMinimumBalance();
    } else {
        System.out.println("minimum balance
check can only be applied to current
account.");
    }
    break;
```

```
            case 6 :
                System.out.println("Exiting program ");
                running = false;
                break;
            default :
                System.out.println("Invalid!");

        }
    }
    scanner.close();
}
}
```

```
Enter customer name : Sameksha
Enter accountType (1 for savings, 2 for current):
1
Enter account number : 2345SE59
Enter initial deposit for savings account : 5000
Bank Operations :
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest
5. Check and apply balance penalty
6. Exit
Enter your choice :
1
Enter deposit amount : 3000
Deposited ! current balance : 8000.0
```

Me

Me

2
Enter withdrawal amount : 4000
withdrawal successful! balance : 4000.0

Me

3
Account balance : 4000.0

Me
4
Interest of 160.0 has been added!
New balance : 4160.0

Me
5
minimum balance check can only be applied to current account.

Me
2
Enter withdrawal amount : 5000
Insufficient balance! withdrawal failed!

Me
6.
Exited!

=> Enter initial deposit for current account : 50000
Me
1
Enter deposit amount : 100
Deposited ! current balance : 50100.0
Me
2
Enter withdrawal amount : 50000
withdrawal successfull ! current balance : 100

Me
5

Balance is below minimum. A penalty has been charged / deducted. New balance : 50

Me *
6
Exited !

21/11/24