

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. 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            System.out.println("Amount withdrawn: " + amount);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 :

```

PS C:\Users\satis\OneDrive\Documents\oop_lab> javac Bank.java
PS C:\Users\satis\OneDrive\Documents\oop_lab> java Bank
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

```

```
PS C:\Users\satis\OneDrive\Documents\oopj_lab> javac Bank.java
PS C:\Users\satis\OneDrive\Documents\oopj_lab> java Bank
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
```

Written Code & Output :

WEEK-5
7

CLASSMATE
Date _____
Page _____

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 withdrawl 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 Rs 10 is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes curr-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 withdrawl 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;  
    int accountNumber;  
    String accountType;  
    double balance;
```

```

public Account (String customerName, int accountNum
    - ber, 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 SavingsAcct extends Account {
    private double interestRate;
    public SavingsAcct (String customerName, int
        accountNumber, double interestRate) {
        super (customerName, accountNumber);
        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;
        S.O.P ("Amount withdrawn:" + amount);
        S.O.P ("Updated balance:" + balance);
    } else {
        S.O.P ("Insufficient balance");
    }
}

```

~~class CurrentAcct extends Account {~~

```

double minimumBalance;
double serviceCharge;

public CurrentAcct (String customerName, int
accountNumber, double minimumBalance, double
serviceCharge) {
    super (customerName, accountNumber, "Current");
    this.minimumBalance = minimumBalance;
    this.serviceCharge = serviceCharge;
}

```

~~super (customerName, accountNumber, "Current");~~

```

public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        S.O.P ("Amount withdrawn:" + amount);
    } else if (balance < minimumBalance) {
        imposePenalty();
        S.O.P ("Updated balance:" + balance);
    }
}

```

```
else {  
    "Insufficient balance");  
}  
}
```

```
private void imposePenalty(){  
    balance = serviceCharge;
```

```
    S.O.P ("Balance fell below minimum service  
charge imposed: " + serviceCharge);  
}
```

```
public class Bank {  
    public static void main (String [] args) {  
        Scanner sc = new Scanner (System.in);  
        S.O.P ("Choose account type: ");  
        Account (n 1. Savings  
        Account (n 2. Current Account");  
        int choice = scanner.nextInt();  
        (sc.nextLine());
```

```
        S.O.P ("Enter customer name: ");  
        string name = sc.nextLine();
```

```
        S.O.P ("Enter account number: ");  
        int accNum = sc.nextInt();
```

```
if (choice == 1) {
```

```
    S.O.P ("Enter interest rate for savings account: ");  
    double interestRate = sc.nextDouble();
```

```
    SavingsAcct savAccount = new SavingsAcct (name,  
    accNum, interestRate);
```

```
    S.O.P ("Enter amount to deposit: ");
```

```
    double deposit = sc.nextDouble();
```

```
scvAccount. computeAndDepositInterest();  
s.o.p ("Enter amount to withdraw:");  
double withdrawAmount = sc.nextDouble();
```

```
scvAccount.withdraw(withdrawAmount);
```

```
{ else if (choice == 2) {  
    s.o.p ("Enter minimum balance for current  
    account:"); minBalance = sc.nextDouble();  
    s.o.p ("Enter service charge for falling below  
    minimum balance:");  
    double serviceCharge = sc.nextDouble();
```

```
CurrentAcct curAccount = new CurrentAcct (name,  
    accNum, minBalance, serviceCharge);
```

```
s.o.p ("Enter amount to deposit:");  
double deposit = sc.nextDouble();  
curAccount.deposit(deposit);
```

```
s.o.p ("Enter amount to withdraw:");  
double withdrawAmount = sc.nextDouble();
```

```
curAccount.withdraw(withdrawAmount);
```

```
{ else if (choice == 3) {  
    s.o.p ("Invalid account type selected.");  
}
```

```
sc.close();
```

```
{ } // End of switch statement
```

```
{ } // End of program
```

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: 3.1.

Enter amount to deposit: 5000.0

Updated balance: 5000.0

Interest added: 150.0

Updated balance: 5150.0

Enter amount to withdraw: 4800.0

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

28/11/21
Enter minimum balance for current account: 1000

Enter service charge for falling below minimum balance:

150

Enter amount to deposit: 6000.0

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