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            System.out.println("Amount withdrawn: " + amount);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

```
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
```

Lab 5

Develop a Java program to create Bank account that maintains two kinds of account for its customers, one called savings & current. Saving account provides CI and withdrawls but no cheque book. The current account provides cheque book but no interst. Current account holders should also maintain a min balance and if balance falls below service charge is imposed. Create a class Account that stores account number and type of account from this derive Cur-acct & Sav-acct

a) Accept deposit from customer & update balance. Display balance
b) Display the balance
c) Compute and depait interst
d) Permit withdrawl and updat balance
Check for the minimum balance, impose penality if necessary

```
import Java.util.Scanner

abstract class Account{
    public String customername;
    public String accountNumber;
    public String accountType;
    public double balance;
    public Account(String c_name, String a_num, String a_Type, double d){
        customername = c_name;
        accountNumber = a_num;
        accountType = a_Type;
        balance = d;
    }
    public void deposit(double amount){
        if(amount > 0){
            balance += amount;
            System.out.println("Deposited");}
        else{
            System.out.println("Deposit must be positive");
        }
    }
    public abstract void compoundInterst();
    public abstract void applyPenalities();
```

```
public void withdraw(double amount){
    if(amount> && amount <= balance){
        balance -= amount;
        System.out.println("withdrawn" + amount);
    } else{
        System.out.println("Insufficient balance or invalid");
    }
}
public void displaybalance(){
    System.out.println("Account holder:" + customer name);
    System.out.println("Account number:" + account number);
    System.out.println("Account type:" + accountType);
    System.out.println("current balance:" + balance);
public double get balance(){
    return balance;
}
}
class currAcct extends Account{
    private double min balance;
    privale double servicecharge;
public currAcct(String c_name, String a_num, double i_bal, double minbal)
    Super(c_name, a_num, "current Account", i_bal);
    minBalance = minbal;
    servicecharge = 50;
}
public void comput interest(){
    System.out.println("No interst for current accounts.");
}
public void applypenalty(){
    if(balance < minbalance){
        balance -= servicecharge;
        System.out.println("Penalty of" + Service charge + "applied");
    }
}
```

```
class SavAcct extends Account{
    private double Ann_Intrate;
    public SavAcct(String c_name, String a_num, double i_bal, double ai){
        super(c_name, a_num, "savings account", i_bal)
        Ann_Intrate = ai;
    }
    public void computeInterest(){
        double interest= balance * Math.pow(1+Annual Intrate/100,1)-balance
        balance= interest
        System.out.println("Interest of "+interest+" applied");
    }
    public void penalty(){
        System.out.println("No penalty for savings account");
    }
}
public class Bank{
    public static void main (String args[]){
        Scanner sc = new Scanner(System.in)
        System.out.println("Enter details for current Account
        System.out.println("Customername, Accountnumber, Intial.
                            minbalance");
        String ~~cutome~~
            c_name= scanner.nextLine();
        String a_num = scanner.nextline();
        double i_bal = scanner.nextdouble();
        double min_bal = scanner.nextdouble();
    Account current = new CurrAcct(c_name, a_name, i_bal, min_ba
    System.out.println("Enter details for savings Account"
    c_name = scanner.nextLine();
    a_num = scanner.nextline();
    i_bal = scanner.nextdouble();
    ~~Min_bal~~ = scanner.nextdouble();
        Ann_irate
    Account saving = new SavAcct(c_name, a_num, i_bal, ~~Mi~~
                                                        An
```