**//Example 1: Demonstration\_1**

/\* The following Java application shows how the transactions in a bank can be carried out concurrently. \*/

class Account {

public int balance;

public int accountNo;

void displayBalance() {

System.out.println("Account No:" + accountNo + "Balance: " + balance);

}

**synchronized void deposit(int amount)**{

balance = balance + amount;

System.out.println( amount + " is deposited");

displayBalance();

}

**synchronized void withdraw(int amount){**

balance = balance - amount;

System.out.println( amount + " is withdrawn");

displayBalance();

}

}

class TransactionDeposit implements Runnable{

int amount;

Account accountX;

TransactionDeposit(Account x, int amount){

accountX = x;

this.amount = amount;

new Thread(this).start();

}

public void run(){

accountX.deposit(amount);

}

}

class TransactionWithdraw implements Runnable{

int amount;

Account accountY;

TransactionWithdraw(Account y, int amount) {

accountY = y;

this.amount = amount;

new Thread(this).start();

}

public void run(){

accountY.withdraw(amount);

}

}

class Demonstration\_1{

public static void main(String args[]) {

Account ABC = new Account();

ABC.balance = 1000;

ABC.accountNo = 111;

TransactionDeposit t1;

TransactionWithdraw t2;

t1 = new TransactionDeposit(ABC, 500);

t2 = new TransactionWithdraw(ABC,900);

}

}

**Output:**

500 is deposited

Account No:111Balance: 1500

900 is withdrawn

Account No:111Balance: 600