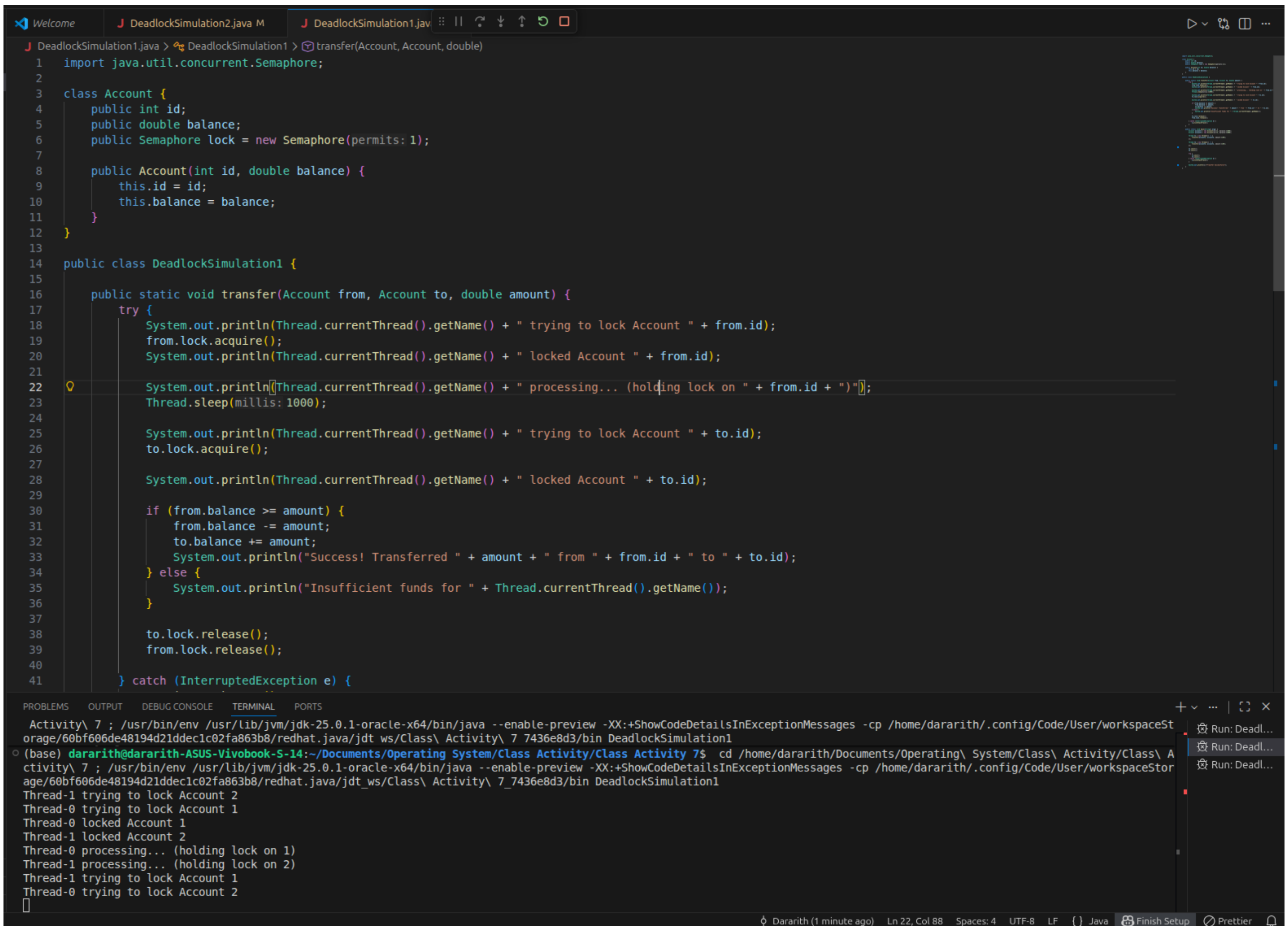
Report: Deadlock Simulation & Solution

Part 1: Deadlock Simulation

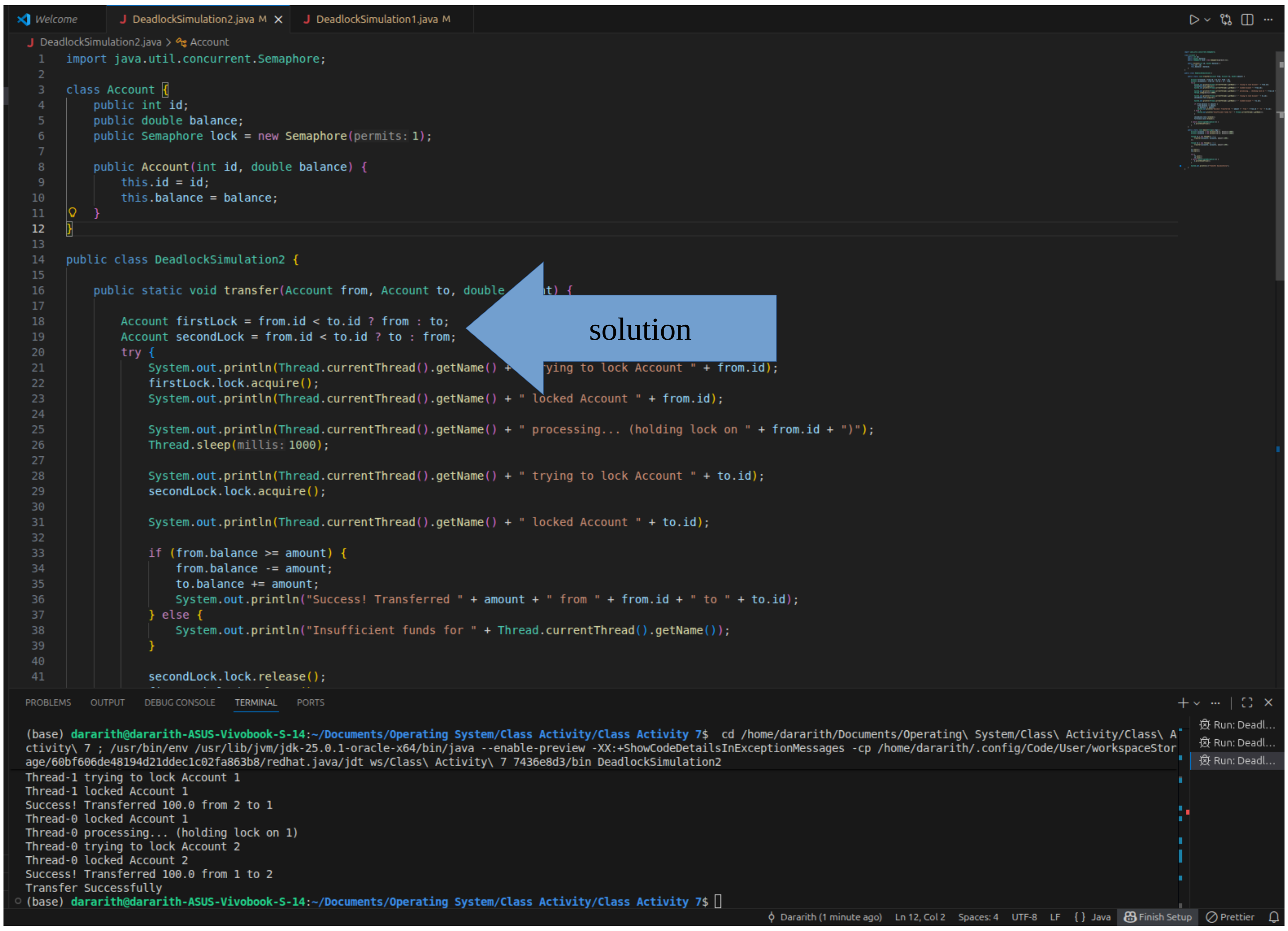


```java
import java.util.concurrent.Semaphore;

class Account {
    public int id;
    public double balance;
    public Semaphore lock = new Semaphore(permits: 1);

    public Account(int id, double balance) {
        this.id = id;
        this.balance = balance;
    }
}

public class DeadlockSimulation1 {

    public static void transfer(Account from, Account to, double amount) {
        try {
            System.out.println(Thread.currentThread().getName() + " trying to lock Account " + from.id);
            from.lock.acquire();
            System.out.println(Thread.currentThread().getName() + " locked Account " + from.id);

            System.out.println(Thread.currentThread().getName() + " processing... (holding lock on " + from.id + ")");
            Thread.sleep(millis: 1000);

            System.out.println(Thread.currentThread().getName() + " trying to lock Account " + to.id);
            to.lock.acquire();

            System.out.println(Thread.currentThread().getName() + " locked Account " + to.id);

            if (from.balance >= amount) {
                from.balance -= amount;
                to.balance += amount;
                System.out.println("Success! Transferred " + amount + " from " + from.id + " to " + to.id);
            } else {
                System.out.println("Insufficient funds for " + Thread.currentThread().getName());
            }

            to.lock.release();
            from.lock.release();

        } catch (InterruptedException e) {
```

```
 Activity\ 7 ; /usr/bin/env /usr/lib/jvm/jdk-25.0.1-oracle-x64/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /home/dararith/.config/Code/User/workspaceSt
orage/60bf606de48194d21ddec1c02fa863b8/redhat.java/jdt ws/Class\ Activity\ 7 7436e8d3/bin DeadlockSimulation1
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$  cd /home/dararith/Documents/Operating\ System/Class\ Activity/Class\ A
ctivity\ 7 ; /usr/bin/env /usr/lib/jvm/jdk-25.0.1-oracle-x64/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /home/dararith/.config/Code/User/workspaceStor
age/60bf606de48194d21ddec1c02fa863b8/redhat.java/jdt_ws/Class\ Activity\ 7_7436e8d3/bin DeadlockSimulation1
Thread-1 trying to lock Account 2
Thread-0 trying to lock Account 1
Thread-0 locked Account 1
Thread-1 locked Account 2
Thread-0 processing... (holding lock on 1)
Thread-1 processing... (holding lock on 2)
Thread-1 trying to lock Account 1
Thread-0 trying to lock Account 2
```

Welcome
DeadlockSimulation2.java M
DeadlockSimulation1.java
DeadlockSimulation1.java > DeadlockSimulation1 > transfer(Account, Account, double)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Run: Deadl...
Dararith (1 minute ago) Ln 22, Col 88 Spaces: 4 UTF-8 LF { } Java Finish Setup Prettier

Part 2: Deadlock Solution (Lock Ordering)



```java
import java.util.concurrent.Semaphore;

class Account {
    public int id;
    public double balance;
    public Semaphore lock = new Semaphore(permits: 1);

    public Account(int id, double balance) {
        this.id = id;
        this.balance = balance;
    }
}

public class DeadlockSimulation2 {

    public static void transfer(Account from, Account to, double          t) {

        Account firstLock = from.id < to.id ? from : to;
        Account secondLock = from.id < to.id ? to : from;
        try {
            System.out.println(Thread.currentThread().getName() +       rying to lock Account " + from.id);
            firstLock.lock.acquire();
            System.out.println(Thread.currentThread().getName() + " locked Account " + from.id);

            System.out.println(Thread.currentThread().getName() + " processing... (holding lock on " + from.id + ")");
            Thread.sleep(millis: 1000);

            System.out.println(Thread.currentThread().getName() + " trying to lock Account " + to.id);
            secondLock.lock.acquire();

            System.out.println(Thread.currentThread().getName() + " locked Account " + to.id);

            if (from.balance >= amount) {
                from.balance -= amount;
                to.balance += amount;
                System.out.println("Success! Transferred " + amount + " from " + from.id + " to " + to.id);
            } else {
                System.out.println("Insufficient funds for " + Thread.currentThread().getName());
            }

            secondLock.lock.release();
```

```
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$  cd /home/dararith/Documents/Operating\ System/Class\ Activity/Class\ A
ctivity\ 7 ; /usr/bin/env /usr/lib/jvm/jdk-25.0.1-oracle-x64/bin/java --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp /home/dararith/.config/Code/User/workspaceStor
age/60bf606de48194d21ddec1c02fa863b8/redhat.java/jdt ws/Class\ Activity\ 7 7436e8d3/bin DeadlockSimulation2
Thread-1 trying to lock Account 1
Thread-1 locked Account 1
Success! Transferred 100.0 from 2 to 1
Thread-0 locked Account 1
Thread-0 processing... (holding lock on 1)
Thread-0 trying to lock Account 2
Thread-0 locked Account 2
Success! Transferred 100.0 from 1 to 2
Transfer Successfully
(base) dararith@dararith-ASUS-Vivobook-S-14:~/Documents/Operating System/Class Activity/Class Activity 7$
```