**Lab 5: Deadlocks**

**Assignment 3 (20 points)** – Programing Problem 8.33

Please show disadvantages the following solution.

**1.** **Deadlock Issue**:

• In this solution, if two transactions simultaneously try to perform a transaction between two different accounts, a deadlock can occur. For example, one thread may lock Account A while another locks Account B, and both threads are waiting for each other to release their locks, causing the system to freeze.

**2.** **Lack of Lock Ordering**:

• Although the solution currently uses synchronized to handle synchronization, there is no control over the order in which accounts are locked. This can lead to situations where accounts are not locked in a consistent order, increasing the likelihood of a deadlock.

**3.** **Inefficient Solution**:

• Using synchronized in the transaction() method can cause system bottlenecks when many transactions happen concurrently. Synchronizing the entire method reduces system performance because only one thread can perform a transaction at a time.

**4.** **Scalability Issues**:

• The current solution using synchronized is more suitable for a limited number of simultaneous transactions. When the volume of transactions increases, such synchronization can cause bottlenecks, affecting the performance and scalability of the system.

**5.** **Issues with identityHashCode()**:

• Using System.identityHashCode() to determine lock order may not always be reliable. Accounts could be locked in an inconsistent or undesirable order, leading to potential deadlocks unless a proper strategy is applied.