Write a multithreaded Java program that uses either semaphores or Java synchronization to control access to a one-lane bridge. One thread should simulate eastbound vehicles and another thread should simulate westbound vehicles. The vehicles do not have a reverse gear, so the bridge would become deadlocked if both an eastbound vehicle and a westbound vehicle were allowed to drive onto the bridge at the same time. Therefore, mutual exclusion must be enforced. Your solution should avoid both deadlock and starvation (e.g. the bridge being monopolized by westbound vehicles while the eastbound vehicles never get to cross). Vehicles traveling in either direction should wait (sleep) for some amount of time, then attempt to cross. Once a vehicle is on the bridge, it should sleep for some amount of time to simulate how long it takes to drive across the bridge. Output a message when each vehicle drives onto the bridge and another message when that vehicle has completed the crossing. Simulate several vehicles traveling in each direction.