|  |  |
| --- | --- |
| **GDPARCM Lecture – Process Sync using Monitors and Semaphores** | Instructor: Neil Patrick Del Gallego |

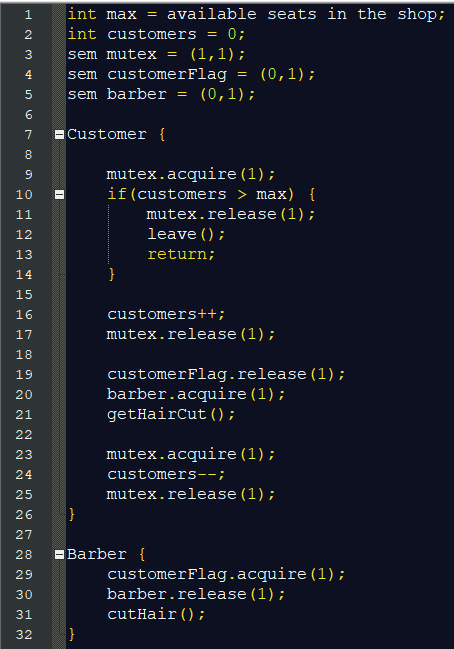
This lesson serves as a culminating activity for monitors and semaphores.

**Activities:** Solve the following problems.

1. A barbershop consists of a waiting room with n chairs, and the barber room containing the barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy, but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers.
   * Customer threads should invoke a function named **getHairCut**.
   * If a customer thread arrives when the shop is full, it can invoke **leave**, which does not return.
   * Barber threads should invoke **cutHair**.
   * When the barber invokes **cutHair** there should be exactly one thread invoking **getHairCut** concurrently.
2. Solve the barber shop problem using **semaphores**.
3. Solve the barber shop problem using **monitors.**

|  |  |
| --- | --- |
|  | |
|  |  |

**SOLUTION for A:**



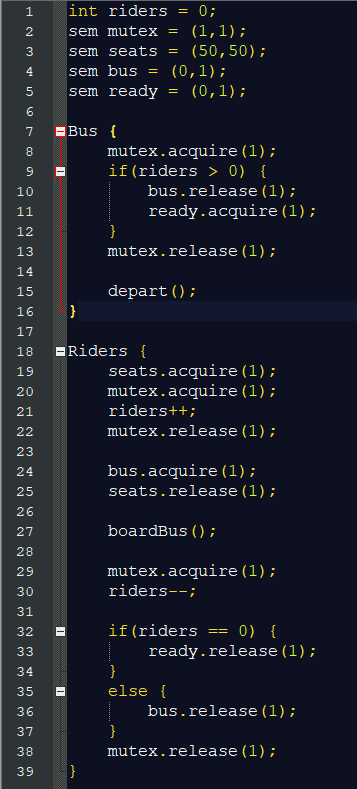
**SOLUTION for B:**

|  |  |
| --- | --- |
|  |  |

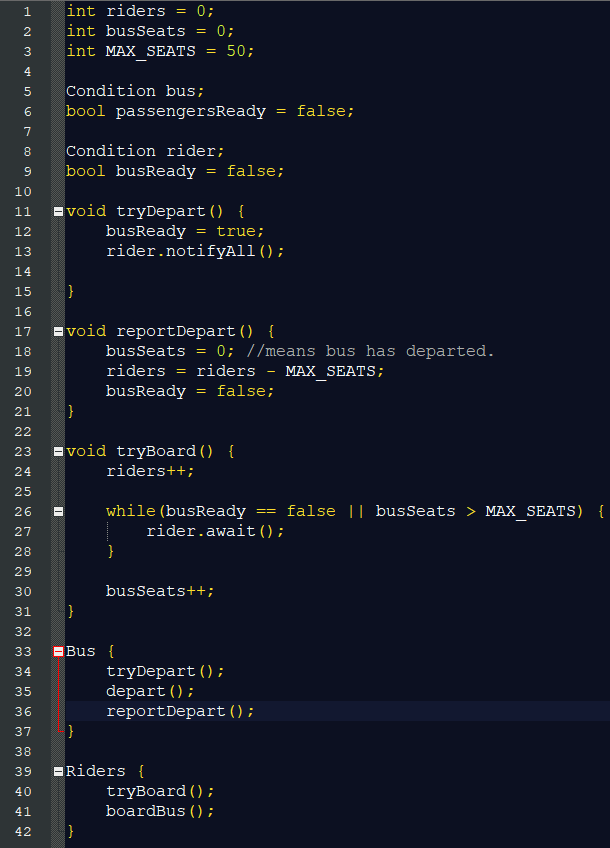
1. Riders come to a bus stop and wait for a bus. When the bus arrives, all the waiting riders invoke **boardBus**, but anyone who arrives while the bus is boarding must wait for the next bus. The capacity of the bus is 50 people; if there are more than 50 people waiting, some will have to wait for the next bus. When all the waiting riders have boarded, the bus can invoke **depart**. If the bus arrives when there are no riders, it should **depart** immediately.
   * 1. Solve the bus problem using **semaphores.**
     2. Solve the bus problem using **monitors.**

|  |  |
| --- | --- |
|  | |
|  |  |

**SOLUTION for A:**

****

**SOLUTION for B:**

****