

Assignment 2 – Synchronization

Submission Rules:

- You must submit **only one “.java” file** containing the source code.
- The submitted file name must follow this format:
ID1_ID2_ID3_Group (e.g. 20190000_20190001_20190002_G1_G2.java).
- The assignment is submitted in groups of **maximum 4** students.
- The deadline for submitting the solution is **16/11/2023 @11:59 PM**.
- Team Members **MUST** be within the same group, or with same TA.
- **Cheating** is totally prohibited and won't be tolerated (any similarity between your code and any other source will be assigned **NEGATIVE Five** without argument).
- **No submissions after the deadline.**

Problem Definition:

It is required to simulate a limited number of devices connected to a router's Wi-Fi using Java threading and semaphore. Routers can be designed to limit the number of open connections. For example, a Router may wish to have only N connections at any point in time. As soon as N connections are made, the Router will not accept other incoming connections until an existing connection is released. Explain how semaphores can be used by a Router to limit the number of concurrent connections.



Following rules should be applied:

- The Wi-Fi number of connected devices is initially empty.
- If a client is logged in (print a message that a client has logged in) and if it can be served (means that it can reach the internet), then the client should perform the following activities:
 - ◆ **Connect**
 - ◆ **Perform online activity**
 - ◆ **Log out**

Note: these actions will be represented by printed messages, such that there is a random waiting time between the printed messages when a client connects, do some online activities and logged out.

- If a client arrives and all connections are occupied, it must wait until one of the currently available clients finish his service and leave.
- After a client finishes his service, he leave and one of the waiting clients (if exist) will connect to the internet.

Solution Design:

Your program must contain the following classes:

1. **Router Class:** This class contains a list of connections and methods to occupy a connection and release a connection.
2. **Semaphore Class:** As given in the synchronization lab (Check Lab 3).
3. **Device Class:** Represent different devices (threads) that can be connected to the router;
Each device has its own name (i.e. C1) and type (i.e. mobile, pc, tablet...) and it may perform three activities: **connect, perform online activity and disconnect/logout.**
4. **Network Class:** This class contains the main method in which the user is asked for two inputs:
 - **N:** max number of connections a router can accept
 - **TC:** total number of devices that wish to connect).
 - **TC lines** that contain: **Name of each device**, and its **Type**.

Program Output:

You will print the output logs in a file, which simulates the execution order of the devices threads and the printed messages of each device.

NOTE THAT: This is just an example not the only scenario that can be applied.

Sample Input:

What is the number of WI-FI Connections?

2

What is the number of devices Clients want to connect?

4

C1 mobile

C2 tablet

C3 pc

C4 pc

Sample Output:

- (C1) (mobile) arrived
- (C2) (tablet) arrived
- Connection 1: C1 Occupied
- Connection 2: C2 Occupied
- C4 (pc) arrived and waiting
- C3 (pc) arrived and waiting
- Connection 1: C1 login
- Connection 1: C1 performs online activity
- Connection 2: C2 login
- Connection 2: C2 performs online activity
- Connection 1: C1 Logged out
- Connection 1: C4 Occupied
- Connection 1: C4 log in
- Connection 1: C4 performs online activity
- Connection 2: C2 Logged out
- Connection 2: C3 Occupied

Grading Criteria:

Router Class	10
Semaphore Class	5
Device Class	10
Network Class	10
Output Valid Order (Code Run Properly)	15

Total : 50 marks will be scaled to 5.