Yusuf Fawzy Elnady

Java Synchronization & Threading

Summary:

This program simulates a petrol station program using **Java threading** and **semaphore**.

Description

The petrol station has n pumps; **only one client** can be served on a **single pump** at a time validating the following rules:

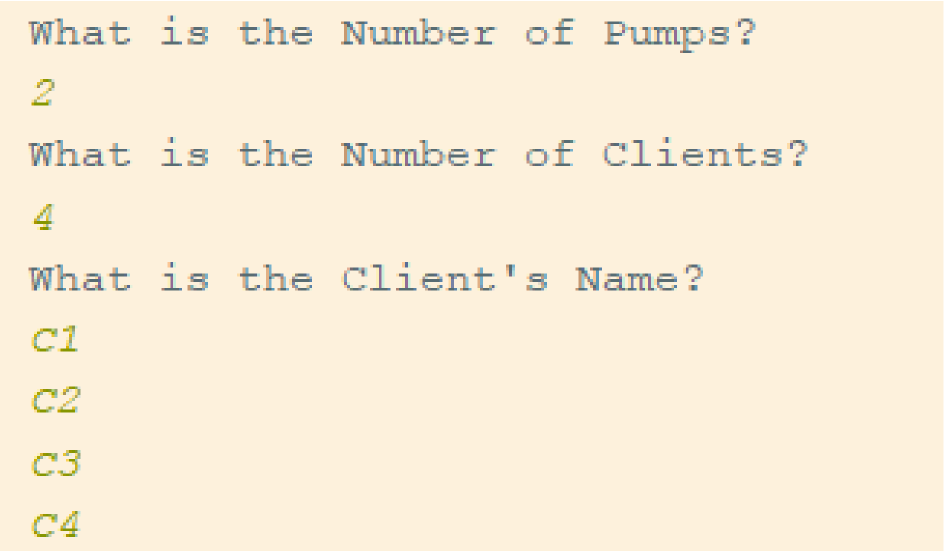
1- The petrol station is initially empty.

2- If a client arrives (prints a message that a client has arrived) and if a free pump exists, the client should (Occupy pump → Get served → Pay → Leave).

Note: The actions are represented by printed messages, such that there is a random waiting time between the printed messages when a client arrives, gets served, pays and leaves.

Program Input:

N: Number of pumps inside the petrol station.
TC: total number of clients with their names (ex: C1, C2, C3…).

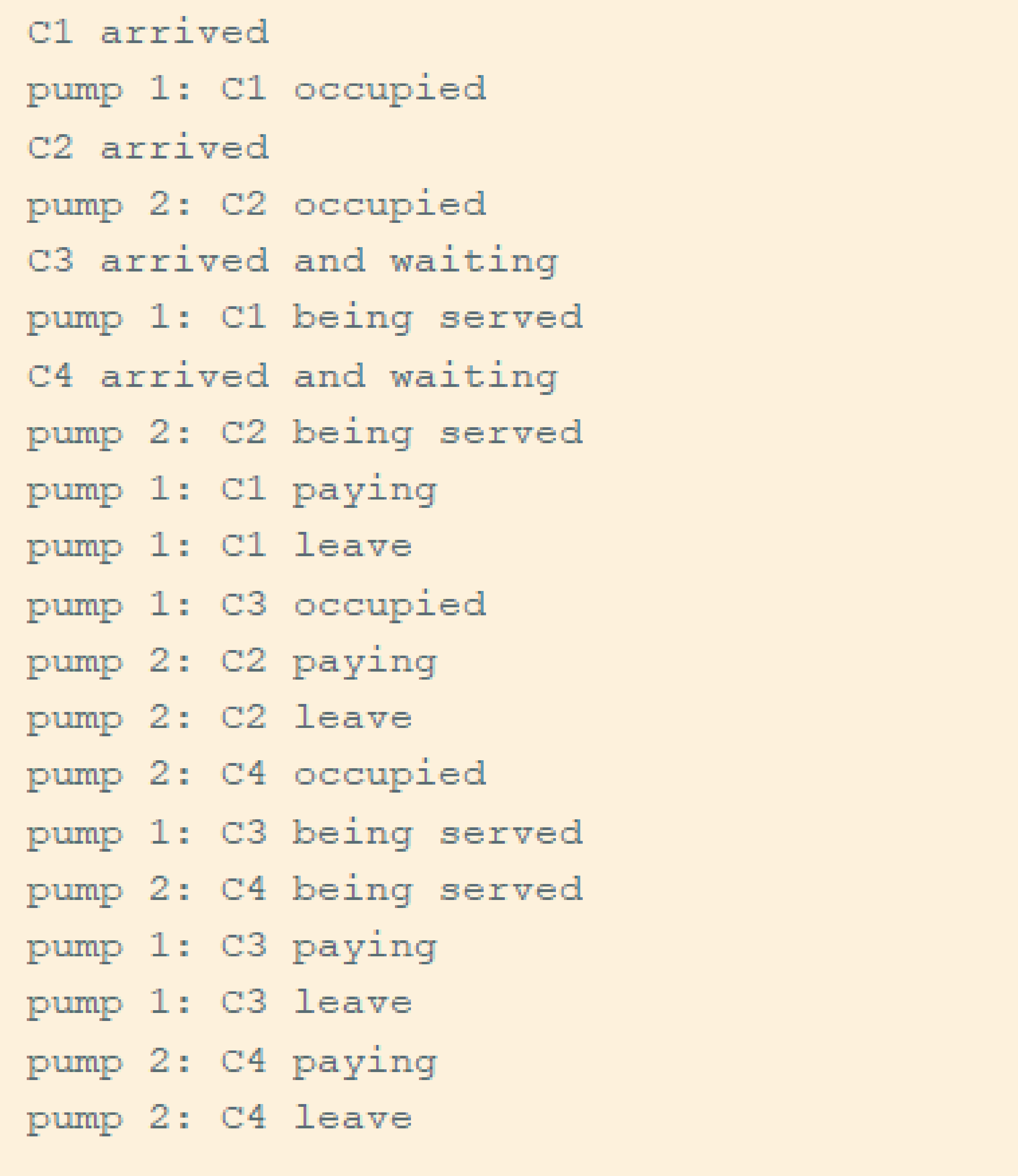


```
What is the Number of Pumps?
2
What is the Number of Clients?
4
What is the Client's Name?
C1
C2
C3
C4
```

Program Output:

The execution order of the Clients’ threads and the printed messages of each client.

Yusuf Fawzy Elnady



```
C1 arrived
pump 1: C1 occupied
C2 arrived
pump 2: C2 occupied
C3 arrived and waiting
pump 1: C1 being served
C4 arrived and waiting
pump 2: C2 being served
pump 1: C1 paying
pump 1: C1 leave
pump 1: C3 occupied
pump 2: C2 paying
pump 2: C2 leave
pump 2: C4 occupied
pump 1: C3 being served
pump 2: C4 being served
pump 1: C3 paying
pump 1: C3 leave
pump 2: C4 paying
pump 2: C4 leave
```