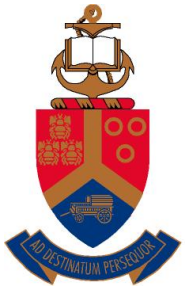

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
COS 226 - Concurrent Systems Date Issued: 15 September 2021

Assignment 1

- **Due Date:** 30 September 2021
- **Assessment:** The assignment will be assessed via live demo.
- This assignment consists of 1 task. Read the task **carefully**!

1 Information

1.1 Objective

This assignment aims to test all the concepts that have been learned so far regarding mutual exclusion and locking via practical implementation without assistance.

1.2 Provided Code

A basic class setup for the simulation has been provided. You will have to create some classes from scratch.

1.3 Mark Allocation

For the assignment, in order to achieve any marks, the following must hold:

- Your code must produce console output. (As this is not marked by fitchfork, formatting is not that strict)
- Your code must not contain any errors. (No exceptions must be thrown)
- Your code may not use any external libraries.
- Your name and student number MUST appear in EVERY file you upload.

The mark allocation is as follows:

Task Number	**Marks**
Implementation	10
Explanation of Code	10
Total	20

2 Assignment

For this assignment you are tasked with simulating a day at a car wash.

2.1 Car Wash

The car wash you are simulating washes and dries cars separately. Both the wash station and the dry station have a queue for cars currently waiting to be washed or dried. The queue of cars to be washed has been set up for you as well as the time it will take to wash and dry each car respectively.

2.2 Employees

The car wash has two types of jobs available, washers and driers.

- A washer may only wash cars and a drier may only dry cars.
- Each employee will work for a random amount of time between 100ms and 500ms before they require a break.
- The break must be a random amount of time between 50 and 100ms.
- Employees will work until all cars are washed and dried.
- The company employs a variable amount of washers and driers.
- Only one washer may be washing a car at a given time, likewise only one dryer may be drying a car at a time.
- A washer and a drier may work at the same time.
- After a car has been washed, it must be added to the 'drying' queue.

2.3 Output

The following output must be produced:

- When an employee is ready to wash/dry a car (they aren't on break):
  [Thread-Name] is ready to wash/dry a car.
- If an employee finishes a car during their shift:
  [Thread-Name] finished washing/drying [Car-Name]
- If an employee doesn't finish washing/drying a car:
  [Thread-Name] washed [Car-Name] for [Time] ms. Time remaining: [Time-Remaining on car]
- When an employee goes on break:
  [Thread-Name] is taking a break.
- Due to the concurrent nature of the program, many outputs may be interleaved, however there are a couple of basic rules to follow:
  - If there has been a 'washing' output, there may not be another one until a 'break' output by the same thread has been output. Likewise with 'drying' outputs.
  - A thread must output 'ready to wash/dry' before their corresponding 'washing/drying' output.

2.4 Example Output

...
Thread-4 is ready to wash a car.
Thread-5 is ready to dry a car.
Thread-2 washed Defender for 167 ms. Time remaining: 1081
Thread-2 is taking a break.
Thread-2 is ready to wash a car.
Thread-3 dried Swift for 281 ms. Time remaining: 383
Thread-3 is taking a break.
Thread-0 washed Defender for 270 ms. Time remaining: 811
Thread-0 is taking a break.
Thread-3 is ready to dry a car.
Thread-0 is ready to wash a car.
Thread-1 finished drying Swift
Thread-4 washed Defender for 431 ms. Time remaining: 380
Thread-4 is taking a break.
Thread-4 is ready to wash a car.
Thread-2 finished washing Defender
Thread-2 is taking a break.
...

2.5 Notes

- You will have to get creative with handling threads, queues and locks.
- A general rule of thumb is that each queue should have their OWN lock when accessed.
- You may not REMOVE any given code.
- You may add as much code and as many classes as you deem necessary.
- Any locks used in the program will have to be written from scratch. i.e NO using java's pre-built locks.
- You may use any of Java's pre-built data structures.
- Any locks used must be FAIR.
- Good Luck!