COS226 - Concurrent Systems
Practical 5 Specification

Release Date: 01/09/2025

Due Date: 08/09/2025 at 23:59

Total Marks: 50

1 General Instructions

- Read the entire assignment thoroughly before you start coding.

- This assignment should be completed individually; no group effort is allowed.

- To prevent plagiarism, every submission will be inspected with the help of dedicated software.

- Be ready to upload your assignment well before the deadline, as no extension will be granted.

- If your code does not compile, you will be awarded a mark of zero. Only the output of your program will be considered for marks, but your code may be inspected for the presence or absence of certain prescribed features.

- If your code experiences a runtime error, you will be awarded a zero mark. Runtime errors are considered unsafe programming.

- Ensure your code compiles with Java 8

- The usage of ChatGPT and other AI-Related software is strictly forbidden and will be considered as plagiarism.

2 Plagiarism

The Department of Computer Science considers plagiarism a serious offence. Disciplinary action will be taken against students who commit plagiarism. Plagiarism includes copying someone else's work without consent, copying a friend's work (even with consent), and copying material (such as text or program code) from the Internet. Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to http://www.library.up.ac.za/plagiarism/index.htm (from the main page of the University of Pretoria site, follow the Library quick link, and then choose the Plagiarism option under the Services menu). If you have any form of question regarding this, please ask one of the lecturers to avoid any misunderstanding. Also note that the OOP principle of code reuse does not mean that you should copy and adapt code to suit your solution.

3 Task (50)

For this task you are responsible for managing a **space station docking port**. Multiple ships (threads) want to dock, but only **one ship can dock at a time**. You are required to implement the docking logic (dock() in Spaceship) using an **Anderson lock** to control access. Once a

spaceship has acquired the lock, it must output "Ship 0 is DOCKING" where 0 indicates the thread id. The thread must then sleep for 200 ms to simulate docking time. Then the spaceship must output "Ship 0 has LEFT the docking port" before unlocking the lock.

Each spaceship has a run() function that must run the dock() function for the number of "dockings" required by the spaceship. In between each dock attempt the spaceship must sleep for a random amount of seconds using: Thread.sleep((int) (Math.random() * 300)) which simulates the ship doing it's spacely duties.

You have been provided with skeleton code.

4 Marking

The marking for this practical will work as follows:

•   You will implement the tasks and upload your solution to Fitchfork.

•   The test classes will be overridden by FitchFork and will then be used to mark the output that is produced by your code.

•   You will receive a mark on Fitchfork.

•   A mark on Fitchfork of greater than 0 will allow you to go to a practical session to be marked by a tutor.

•   In the practical session a tutor will ensure that your implementation has kept within the restrictions (i.e no copy-pasting, not using libraries to do the whole task). The tutor will then assess your understanding of your implementation and the practical content.

•   You will receive a final mark which will be some weighted combination of your Fitchfork mark and your "understanding" mark.

5 Upload Checklist

The following files should be in the root of your archive

•   Main.java will be overridden

•   AndersonLock.java

•   SpaceShip.java

6 Submission

You need to submit your source files on the FitchFork website (https://ff.cs.up.ac.za/). All methods need to be implemented (or at least stubbed) before submission. Place the above-mentioned files in a zip named uXXXXXXXX.zip where XXXXXXXX is your student number. Your code must be able to be compiled with the Java 8 standard.

For this practical, you will have 20 upload opportunities and your best mark will be your final mark. Upload your archive to the appropriate slot on the FitchFork website well before the deadline. **No late submissions will be accepted!**