COS 226 Practical Assignment 2

- Date Issued: Monday, 2 August 2016 13:30h
- Date Due: Friday, 5 August 2016 17:00h
- Submission Procedure: Upload the specified required files for a given task.
- This assignment consists of **2 tasks** with a total of **15 marks** for the entire practical.

---

1. Introduction

From here on forward, the practical assignments assume that you know the basics of how threads work. You must complete this assignment individually.

You may ask the Teaching Assistants for help but they will not be able to give you the solutions. They will be able to help you with coding, debugging and understanding the concepts explained both in the textbook and during lectures.

2. Mark Allocation

This assignment is divided into **two tasks**. Your program must adhere to the following:

1. Your program must produce the expected output
2. Your program must not throw any exceptions
3. Your program must implement the correct mechanisms or algorithms, as specified by the assignment.

3. Source code

Before you begin, you should download the source code from the CS website. You are allowed to make changes to some of the files in the source code.

4. Task 1 – Shared Printer Simulation (5 marks)

In this task, you will have to study the code that is in the *Task1* folder. The files provided are **JobSimulator.java**, **Main.java** and **PrintQueue.java**.

In this task your will have to simulate a shared printer. Different threads present their jobs to be printed by the shared printer. At times the jobs seek to access the shared resource at the same time and there might be a potential risk whereby a document presented by one thread can be printed by and presented to a different thread. You must implementing mutual exclusion using re-entrant locks in order to avoid such occurrences.

The status represents the duration of the print for a particular thread. When a thread is printing for 2 seconds then the other threads must wait for the printer to finish with that job and only then can another job be started. This will continue until all jobs are finished.

Your task is to edit the PrintQueue.java file only and implement the member function *printJob(Object)*. The function must do the following:

- Each thread must sleep for duration of at most 5 seconds. The sleep-time can be created by a random number generator.
- The name of the currently printing thread as well as the time taken to print the document must be displayed into the standard output screen. Here is an example of the print message during printing;

  ```
  Thread-0: PrintQueue: Printing a Job for 2 seconds
  ```

  Where `Thread-0` is the name of the currently printing thread the duration is `2` seconds.

Compress all your files (especially **PrintQueue.java** file) and upload tar-ball (or *.zip* file) to the CS website to the **Practical2 Task1** submission box. Make sure that the uploaded files do not belong to any package.

5. Task 2 – Filter Lock (10 marks)

In this task you must implement the filter lock algorithm that will provides the full implementation of the Lock interface. You are provided with a **Filter.java** file.

The primary methods of the Lock interface which you must implement fully include the *constructor* which takes a single integer argument, the *lock()* function and the *unlock()* function. The other irrelevant methods of the Lock interface may be implemented with default values.

You are expected to test if your functions work appropriately. You must use any lock based implementation of **your choice** to test your filter lock functions. For example you can test your methods by locking any of the previously assigned tasks of the current practical or Practical 1.

Compress the **Filter.java** file and upload the tar-ball to the CS website to the **Practical2 Task2** submission box. Make sure that the uploaded file does not belong to any package.