COS 226 Practical Assignment 5

- Date Issued: Monday, 12th September 2016, 13:30h
- Date Due: Friday, 23rd September 2016, 17:00h
- Submission Procedure: Upload the specified required files for a given task.
- This assignment consists of **1 task** with marks stated alongside each question.

---

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 **one (1) task**. Each task will be marked by a Teaching Assistants manually. Please upload your code to the provided upload slots. 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 – Consensus Protocol (10 marks)

Suppose we augment the FIFO Queue class with a peek() method that returns but does not remove the first element in the queue. Implement a consensus protocol that makes use of a queue with a peek() method in order to show that a solution may be obtained for any number of threads (i.e. that a queue with a peek() method has an infinite consensus number). You are given a class, Executor, with a program entry point, a class. Consensus, that defines the interface of a consensus protocol implementation, a class, ConsensusProtocol, which is the base class for all consensus protocol implementations, and an incomplete ConsensusThread, that defines a thread that executes the consensus protocol, an interface as well as an incomplete consensus protocol implementation, PeekConsensus.

The program must execute as follows: The executor creates a specified number of threads. All but one of the threads are delayed for a random duration of time. All of the threads will invoke the decide() method of your consensus protocol exactly once. Before calling the decide() method, each thread writes out its identifier, input value, and sleep duration. After calling the decide() method, each thread will write out its identifier and the decision value that it obtained. In addition to these outputs, you must print out the following lines: 1. The thread id and the value written for each call (if any) to your queue's enqueue() method. 2. The thread id and the value observed for each call to your queue's peek() method. Executor constructor takes one argument which is the number of threads to use. It is currently set to 5 but you should test with different values to make sure that it works for all cases.

Compress all your files (especially **PeekConsensus.java and ConsensusThread.java** file) and upload tar-ball (or *.zip* file) to the CS website to the **Practical5** submission box. Make sure that the uploaded files do not belong to any package.