## Concurrent & Parallel Programming

**Question 1**

Ten threads share an integer array that is initialized by main. Each thread when accessing the shared array has exclusive access. This means that no other thread in the group will be accessing it at that time. Hence, locking is not required. The threads agree to access the threads in sequence where the order of the sequence is determined by main. All the threads are released at the same time and may be executed by the scheduler in any order. However, they may only access the data in the sequence pre-determined by main. In effect this means that each thread must wait for earlier threads in the sequence to complete before accessing the data. There are three kinds of thread: TA adds x to each element in the array; TB decrements each element in the array and TC squares each element in the shared sequence. The number of each kind of thread is determined by main.

(Hint: Ask yourself how many Semaphores do I need?)

**Question 2**

A class that forces all threads to wait for an event is required. This class is to have two methods: waitEvent() that forces threads to wait and releaseAll() that releases all waiting threads. The class should be called WaitEventBarrier. It takes no arguments. Write this class using a Semaphore.