## Concurrent & Parallel Programming

**Question 1**

Given on file is a class Point that is thread safe because it is immutable. Your task is to write a class, called SetPoint, that manages a set of Point instances. This class *owns* the points under its control and must synchronize concurrent activity. Hence, the class must be thread safe. The class SetPoint should use a HashSet to store points and must provide the following interface methods: add, that adds a new point to the collection; search, that searches for a point in the class and returns true or false; getAllX(int x) that returns a list of all points whose x-ordinate matches x; toString, that returns a string representation of the elements in the set.

**Question 2**

The file contains a generic class CircularQueue<T> that implements a queue modelled by a circular array. There are two constructors: a default one that sets the size to an arbitrary value of 20 and one that takes the maximum size of the queue as an argument n. In both cases head and tail are set to zero. Both methods join and leave are implemented using modulo arithmetic. Please note that a queue simply manages the order of insertion and removal, it does not interact in any way with the data under its control. This class is not thread safe. Your task is to re-write it so that it is thread safe and solves issues that may arise around iteration.

Hint: This class implements the Iterable interface. Managing iterators is problematic in a concurrent universe and it is important to adopt a strategy to deal with the situation. You may need to do some research to assist in dealing with this problem.