

5 Further Java (arb33)

A concurrent queue allows two or more Java threads to add items to, or remove items from, a shared buffer. Items are stored in the buffer in a strict first-in-first-out order. The buffer in this question has a fixed size and therefore once the buffer is full, any attempt to add items to the buffer should block until enough items are removed that there is space to store the additional items. If the buffer is empty, any attempt to remove items from the buffer should block until an item becomes available.

A novice Java programmer implements a concurrent queue as follows:

```
public class IncorrectConcurrentFixedSizeQueue {
    private int[] buffer = new int[10];
    private int front, back, count;
    public void put(int val) throws Exception {
        while (count++ >= 10)
            Thread.sleep(1000);
        buffer[front++ % 10] = val;
    }
    public int get() {
        count--;
        return buffer[back++ % 10];
    }
}
```

- (a) Describe the intent of the variables `buffer`, `count`, `front` and `back`. [4 marks]
- (b) Describe, with justification, three problems with the above implementation which means it does not adhere to the specification of a concurrent queue. [6 marks]
- (c) Write a new implementation of a *generic* concurrent queue which stores items of type `T`. The constructor should accept an integer to specify the fixed size of the buffer. [8 marks]
- (d) Two Java threads form a pipeline where the output of the first thread is used as an input into the second. A concurrent queue is used to allow the output from the first thread to be used as input into the second. Describe, with justification, a situation where increasing the size of the buffer in the concurrent queue would improve overall system throughput. [2 marks]