

4F14 Coursework: task

The code should consist of exactly **one** C++ file with the name `4f14_crsid.cc` (where “crsid” is your CRSid, e.g. “pok21”). No external libraries beyond the C++ standard library is allowed. Your code must compile correctly running the following command on a modern GCC compiler:

```
g++ -Wall --std=c++11 4f14_crsid.cc -o 4f14
```

- a) Implement a queue data structure that holds items that can be added and removed from the queue. Each item held in the queue should consist of a single `std::string` and a single integer. The queue should support a means of adding an item to the queue and a means of removing an item from the queue. A queue works as a stack, except for removing an item: in a stack, when an item is removed, the last item added to the stack is removed; in a queue, when an item is removed, the first item added to the queue is removed.
- b) Instantiate your queue and implement a method that populates it by adding 80 items. Each string of each item consisting of 3-7 characters, inclusive (the number of characters in the interval chosen at random), each character being a letter between a and z, inclusive, again chosen at random. Each integer in each item should be a random number between 0 and 255 inclusive. This method should run once in the main function immediately after the queue has been created in the main function of the program.
- c) Start a background thread that runs continually in the background and reverses the ordering of all the items present in the queue. Once all items in the queue have been reversed, the thread should output the sum of all the integer values in the items in the queue that were present when the queue was reversed. This procedure should repeat as long as there are items in the queue. When there are no items in the queue the thread should stop running. You may want to experiment with introducing a small delay to the thread to make it easier to debug.
- d) Start another background thread that runs continually in the background and sequentially prints out all string and integer values for all items currently present in the queue. When there are no more items in the queue, the thread should stop running. You may want to experiment with introducing a small delay to the thread to avoid overloading the console.
- e) Start a third background thread that every 0.2 seconds randomly selects an item, among the total number of items currently in the queue, and then removes the item from the queue. When there are no more items in the queue, the thread should stop running.

4F14 Coursework: assessment

- You will be required to book a 10-minute oral viva examination slot; slots will be available at the start of next term.
- In the oral viva you will be asked to explain how much you have done and the reasoning behind your code and why it works / does not work quite as you intended. It is intended to be a constructive discussion **about your code**, not an examination of your understanding of concurrency in general.
- You will be provided with some immediate feedback at the viva and brief textual feedback via Moodle afterwards.
- It is **EXTREMELY IMPORTANT** you name your file 4f14_crsid.cc, where crsid is your crsid (for example: "4f14_pok21.cc")
- **DO NOT SHARE YOUR CODE WITH ANYONE AND DO NOT SHARE YOUR COURSEWORK CODE ONLINE EITHER BEFORE OR AFTER EXAMINATION OF THIS MODULE**

Deadline and mechanics for handing it in

- The **deadline** is 6th March 2020.
- The assessment must be uploaded via Moodle
- The exact Moodle upload deadline is 6th March 2020 16:00 UK time.