Imperial College
London

TEST FEEDBACK

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

---

# COMP40009
# Computing Practical 1 (Java)
# Section B

---

*Author:*
Dr. Soteris Demetriou

Date: June 5, 2023

# 1 Question 1

Question 1 was worth 5 marks and constitutes 10% of the overall Section B mark. In general most of the students achieved high marks for this question. A common issue here was that some students were not checking if the given input is null.

# 2 Question 2

Question 2 was worth 10 marks and constitutes 20% of the Section B test mark. This could be approached either through an in-place removal approach or through a list traversal to identify the item to remove. Students that followed the simpler but more computationally expensive second approach lost 1 point. There were also some issues with incorrectly rewiring the nodes in the doubly linked list.

# 3 Question 3

Question 3 was worth 10 marks and constitutes 20% of the Section B mark. A common issue here was that some solutions would not move the read item into the most recently used position in the list. Another issue was that some solutions would not utilize the HashMap to access the item in $O(1)$ and instead would try to use `find()` to traverse the list and find the item.

# 4 Question 4

Question 4 was worth 10 marks and constitutes 20% of the Section B mark. One issue was that solutions would not allow overwriting existing items or had some issues with their implementation. Another issue was that some solutions would not remove the LRU item at max capacity. Finally, some solutions would not add the new item correctly to the most recently used position in the list and point to it through the HashMap.

# 5 Question 5

Question 5 was worth 10 marks and constitutes 20% of the Section B mark. Most solutions received all the points by simply implementing the `MemoryI` interface, and reimplementing the methods using the `synchronized` keyword. Some solutions `lock` the method at the beginning, releasing the lock at the end in a `finally` block. Some solutions lost a few points because they would not synchronize all required methods.

# 6 Question 6

Question 6 was worth 5 marks and constitutes 10% of the Section B mark. This was the hardest question as a thread-safe implementation of an LRU cache data structure can be quite complicated. Common issues for losing points was a missing analysis on the LRU concurrency guarantees the proposed solution can give, not using a thread-safe `HashMap`, or not discussing the locking order required between the `append()`, `remove()`, `read()` and `write()` methods, or the locks they need to use.