

COMP108 Data Structures and Algorithms

Lab Exercises (Week 4)

Due: 5 March 2021, 5:00pm

Information

- Submission: Submit the file `COMP108W04.java` to SAM
https://sam.csc.liv.ac.uk/COMP/CW_Submissions.pl?qryAssignment=COMP108-14
- Submission of lab/tutorial exercises contributes to 10% of the overall module mark. Submission is marked on a pass/fail basis - you will get full marks for submitting a *reasonable attempt*.
- Late submission is **NOT** possible. Individual feedback will not be given, but solutions will be posted promptly after the deadline has passed.
- These exercises aim to give you practices on the materials taught during lectures and provide guidance towards assignments.
- Relevant lectures: **Lecture 6 (Video 1) and Lecture 8 (Video 2)**
- You can refer to the guidance on how to use the web-based IDE <https://ide.cs50.io/>.

1. Programming — Preparation

- (a) Download two java files “`COMP108W04App.java`” and “`COMP108W04.java`” from Canvas via the link “Labs & Tutorials” → “Week 4”.
- (b) Compile the programs by typing first **`javac COMP108W04.java`** and then **`javac COMP108W04App.java`**. There should be two files created: `COMP108W04.class` and `COMP108W04App.class`.
- (c) Run the program by typing **`java COMP108W04App`**.
- (d) **Every time you have edited `COMP108W04.java`, you have to (i) recompile by `javac COMP108W04.java` and then (ii) run by `java COMP108W04App`.**

2. The Scenario we work with this week

- (a) **IMPORTANT:** In this lab, we are considering the following scenario. We have a database of movies (stored as integer ID) and some requests from customers who want to watch movies (again stored as integer ID). We want to answer two questions:
 - i. Which requested movies do not exist in the database? Report this in the same order of the requests.
 - ii. For each movie in the database, how many times it has been requested by the customers? Report this in the same order of the database.

We will need to write nested loop to answer these questions.

- (b) You are going to work on `COMP108W04.java` to complete two methods: `notExists()` for Task 1 and `count()` for Task 2.

- (c) For simplicity, the database and requests have been hard-coded in COMP108W04App.java.
 The database contains 70, 20, 60, 40, 50, 30, 10, 80
 The requests contains 5, 10, 60, 70, 15, 50, 30, 20, 20, 20, 25, 15, 20, 10, 20, 5, 70, 70, 10, 10

3. Programming with Nested Loops Task 1

- (a) **Enter your name and student ID** to the beginning of COMP108W04.java.
- (b) As stated above in 2a, we want to look at each movie ID in the requests and report if it does not exist in the database. We can do this by nested loops: the outer loop iterates through the requests one by one and the inner loop iterates through the database one by one and report the movie ID if it does not exist in the database.
Remember: you will need TWO different index variables, e.g., i and j , for the outer and inner loops.
- (c) Implement the method in **notExists()** of COMP108W04.java. This method takes 4 parameters: the first is the array storing the requests and the second is the size of this array; the third is the array storing the database and the fourth is the size of this array.
- (d) **Expected output:** Based on the given database and requests, your method is expected to print out: **5 15 25 15 5**
- (e) What is the time complexity of your algorithm? Justify your answer. Give your answer in the comment section at the beginning of COMP108W04.java.

4. Programming with Nested Loops Task 2

- (a) This time we would like to go through each element of the database and **count** how many times it appears in the requests. We also do this using nested loops and you will have to decide which array the outer and the inner loops each iterate through.
- (b) Implement the method in **count()** of COMP108W04.java. The order of the 4 parameters is the same as Task 1.
- (c) **Expected output:** Your method should print: **3 5 1 0 1 1 4 0**
- (d) What is the time complexity of your algorithm? Justify your answer. Give your answer in the comment section at the beginning of COMP108W04.java.

5. Optional Puzzle for fun [do this only after you finish Tasks 1 & 2, no submission needed]

- (a) Nine eggs look identical except one is heavier. How can you weigh only **two** times on a balance scale to find out which one is heavier? A balance scale can only tell us which side is heavier but it cannot tell by how much. *Hint: If there are three eggs and one is heavier, how to weigh one time only to find out which egg is heavier?*



- (b) What if the different egg may be heavier or lighter (and we don't know if it's heavier or lighter) How can you weigh only **three** times on a balance scale to find out which one is different and whether it is heavier and lighter?