



CentraleSupélec

IS1220

## Tutorial 04

Francesca Bugiotti, Paolo Ballarini

---

### General instructions:

- Ask if you have any problem.
- If you have not done so already, create a workspace for your tutorial IS1220/TDs.
- Within the workspace, create a new package for this TD called `fr.ecp.IS1220.TD04/`.
- Carefully document your code (JavaDoc), i.e., explain the general idea of your algorithm, explain the relevant steps in the code implementing the algorithm, document assumptions (if any), corner cases, and error conditions.

### Main Concepts:

- Java collections
- Reading/writing a file
- Handling exceptions

## Exercise 1. More practice with various Collections

**E1 Q1)** Write a JAVA program that creates your implementation of an `ArrayList` for storing integer values. Insert several values in the list some of which must be duplicates for example let the list be `[2,2,2,2,5,5,8,9,9]`. Write a `removeDuplicate(List l)` method which takes a list as input and remove from the given list all duplicated elements.

**E1 Q2)** Test your method on a list with duplicates such as `[2,2,2,2,5,5,8,9,9]`.

## Exercise 2. Reading/writing from/to files

**E2 Q1)** Write a program that counts the number of occurrences of each word contained in the file `words.txt` and displays the results on screen.

For this you may want to use a **Map** structure (of which kind? mapping what to what?).

**E2 Q2)** Extend your program adding a method that allows you to output the words count result in a file called `wordscount.txt`. Remember that I/O on files require handling of exceptions.

**E2 Q3)** Write a modified version of the above words counting program that counts the words contained in the file `words.txt` line by line. The program at first counts the word for line 1 and adds a line to the file `wordscount.txt`, then counts the words for line 2 and adds a second line to the file, etc. Each line of the file will contain the word count for the corresponding line in the `words.txt` file.

## Exercise 3. Adding exceptions

In this exercise you are required add exception handling to the course marking program of Tutorial 3.

**E3 Q1)** Extend the program you wrote (are writing) as a solution of Exercise 2 of Tutorial 3 by including a control on the weights used for computing the final mark of a given module (starting from the mark splits). The program should control that the weights associated to a module sum up to 1. Thus if WFE (Weight for Final Exam mark), WME (Weight for Midterm Exam mark) and WPW (Weight for Practical Work mark) are the three weights of a module than the program should always check that  $WFE + WME + WPW = 1$  and if not it should rise an exception (thus you should define your own exception).