CSC319 – Java Lambda – Review Exercise

Due date: 6 May 2025, at NOON on CSCMS
Instruction: Work the following problems and zip all your answers into one (1) single *.zip* file for submission.

Q1. Write a Java application in one (1) single file that only employs the JCF to process and return a list of all words from the file, ***the_sign_of_the_four.txt***. Once you have the list of all words, you will sort these words in the list in a descending order using ***the number of characters*** as a criterion (for simplicity, *no need to break the tie*). Then you will print out the ***first occurrence of the longest word*** in the list. Additional requirements are as follows:

- Do **NOT** use wildcard to import classes;
- Your application will read a file called ***the_sign_of_the_four.txt***. The file is provided;
- For simplicity, assume the file name is fixed. Do **NOT** prompt the user for the file name;
- Empty string and spaces must NOT be treated as a word;
- Your application **MUST NOT** use Java lambda/functional interface nor Streams.

Q2. Write a Java application in one (1) single file that modifies the Execute-Around code that we studied in class such that now the **processFile** method will return a list of all words from the file, ***the_sign_of_the_four.txt***. Once you have the list of all words, you will sort these words in the list in a descending order using ***the number of characters*** as a criterion (for simplicity, *no need to break the tie*). Then you will print out the ***first occurrence of the longest word*** in the list. Additional requirements are as follows:

- Do **NOT** use wildcard to import classes;
- Your application will read a file called ***the_sign_of_the_four.txt***. The file is provided;
- For simplicity, assume the file name is fixed. Do **NOT** prompt the user for the file name;
- Empty string and spaces must NOT be treated as a word;
- Your application must include at least one (1) lambda and one (1) method reference;
- Your application **MUST NOT** use Java Streams.

Hint: The **reversed()** method on a lambda expression can be useful for this question.