

## CS1003 Week 4 Exercise 1: Text Processing

As with all lab exercises, this exercise is not assessed. It is intended solely to help you understand the module material. There is probably more here than you will have time for during the lab hour; you are encouraged to complete it in your own time if you don't finish it. The aim of this exercise is to further practise processing of text in files.

### 1. Count the words in a text file

Make a copy of your solution from the second exercise in week 2. Change it so that it just counts the number of words in the input file and prints out the result. If you had a working solution to the week 2 exercise, this should just involve a bit of tidying and removing unneeded code.

Test your program on the test files provided (test-short.txt and test-long.txt).

### 2. Count occurrences of a particular word

Make a copy of your previous program, and modify it so that it counts the number of occurrences of some word specified by the user.

You'll need to change your code so that rather than just counting the number of words in each line of the file, and adding that to the total, it now checks each word to see if it matches the specified word.

### 3. Command-line arguments

If you haven't done so already, modify your program so that it accepts both the name of the file and the word to be counted as command-line arguments (i.e. `args[0]` and `args[1]`) that are passed to the main method).

This way of providing extra information to the program is useful when running it from the terminal. Any extra strings that are typed by the user after the Java command are added to the `args` array, where they can be accessed by the program.

For example, to count the occurrences of the word "Boatswain" you should be able to run your program from the terminal using the command:

```
java Week4Exercise1 test-short.txt Boatswain
```

### 4. Count occurrences of multiple words

Modify your program so that it accepts multiple words to be counted, and records separate counts for each of those words.

You can do this by adapting your solution to part 2 or part 3, and creating an array of strings to contain the words being counted. The easiest way is probably then to create another array of integers to contain the counts for those words. Whenever your program encounters a word in the file, it now needs to go through the string array, comparing the word with each array entry. When it finds a match, it should increment the count in the corresponding slot of the integer array. You'll also need to write a loop at the end to write out all the words being counted, and their individual counts. If all goes well, running your program as shown below should produce the following word counts

```
java Week4Exercise1 test-short.txt Boatswain Master The the
Total number of words in file: 47
Boatswain : 3
Master : 3
The : 0
the : 1
```

**Extension:** Try changing your program to use a `java.util.HashMap` from `String` to `Integer` to record the integer count for each word string (a map permits you to store and retrieve *values* for a given *key*, in this instance the *key* could be the word string and the value the integer count).