# **Practical Lab 07 – Sets and Maps Labs**

## **Lab 1 SpellCheck program**

In this lab you are required to take the program SpellCheck from the VLE along with its related program files and run it and examine the output.

You should then adapt the program so that it is in alphabetical order when outputted.

Tip – use the TreeSet data structure.

## ***Lab 2* Word Frequency**

In this worked example, we read a text file and print a list of all words in the file in alphabetical order, together with a count that indicates how often each word occurred in the file. For example, the following is the beginning of the output that results from processing the book “Alice in Wonderland”:

* a 653
* abide 1
* able 1
* about 97
* above 4
* absence 1
* absurd 2

**Step 1** **Determine how you access the values.**

In our case, the values are the word frequencies. We have a frequency value for every word. That is, we want to use a map that maps words to frequencies.

**Step 2 Determine the element types of keys and values.**

Each word is a String and each frequency is an Integer. (You cannot use an int as a type parameter because it is a primitive type.) Therefore, we need a Map<String, Integer>.

**Step 3 Determine whether element or key order matters.**

We are supposed to print the words in sorted order, so we will want a TreeMap.

**Step 4 If you use a tree, decide whether to supply a comparator.**

The key type for our tree map is String, which implements the Comparable interface. Therefore, we need to do nothing further.

***Here is the pseudocode***

For each word in the input file

Remove non-letters (such as punctuation marks) from the word.

If the word is already present in the frequencies map

Increment the frequency.

Else

Set the frequency to 1.