Workshop 3

All Tasks given below must be submitted via Blackboard submission system by the deadline. Create a zip folder which contains all the **.java** files of the tasks and screen shots for the output.

Task 1

Case study: Occurrences of words

This case study writes a program that counts the occurrences of words in a text and displays the words and their occurrences in alphabetical order of the words.

The program uses a **TreeMap** to store an entry consisting of a word and its count. For each word, check whether it is already a key in the map. If not, add an entry to the map with the word as the key and value **1**. Otherwise, increase the value for the word (key) by **1** in the map. Assume the words are case insensitive; e.g., **Good** is treated the same as **good**.

CountOccurrenceOfWords.java

```
import java.util.*;

public class CountOccurrenceOfWords {
    public static void main(String[] args) {
        // Set text in a string
        String text = "Good morning. Have a good class. " +
                                    "Have a good visit. Have fun!";

        // Create a TreeMap to hold words as key and count as value
        Map<String, Integer> map = new TreeMap<>();

        String[] words = text.split("[ \n\t\r.,;:!?(){}]");
        for (int i = 0; i < words.length; i++) {
            String key = words[i].toLowerCase();

            if (key.length() > 0) {
                if (!map.containsKey(key)) {
                    map.put(key, 1);
                }
                else {
                    int value = map.get(key);
                    value++;
                    map.put(key, value);
                }
            }
```

Introduce → }]

```
        }

        // Get all entries into a set
        Set<Map.Entry<String, Integer>> entrySet = map.entrySet();

        // Get key and value from each entry
        for (Map.Entry<String, Integer> entry: entrySet)
            System.out.println(entry.getValue() + "\t" + entry.getKey());
        }

                OR replace the get all entries with Lambda Expression
        map.foreach((k, v) -> System.out.println(k + "\t" + v));
}
```

Your task is:

Rewrite the example code to read the text from a **text file**. The text file is passed as a command-line argument. Words are delimited by whitespace characters, punctuation marks (**, ; . : ?**), quotation marks (**' "**), and parentheses. Count words in case-insensitive fashion (e.g., consider **Good** and **good** to be the same word). The words must start with a letter.
Display the output in alphabetical order of words, with each word preceded by its occurrence count. Also check the proper Exception.

Task 2

Part a (Easy part)
Generate a lottery of a three digit number. The program prompts the user to enter a three-digit number and determines whether the user wins according to the following rules:

1. If the user input matches the lottery number in the exact order, the award is $10,000.
2. If all digits in the user input match all digits in the lottery number, the award is $3,000.
3. If one digit in the user input matches a digit in the lottery number, the award is $1,000.

Part b

Revise above program to add an additional $2,000 award if two digits from the user input are in the lottery number. Use the Collection's appropriate method to check whether the two digits in the user input are in the lottery number. Proper Exception handling is expected.