Dr. Carsten Gutwenger   Winter 2011/12

Object-oriented Programming
Assignment Sheet No. 4

Date: November 15

Exercise 4.1 (File I/O)

Write a program that reads from two input files `desiredsignal.txt` and `measuredsignal.txt`, which are desired and measured values of a sensor. Compute the error of each measurement, which is the absolute value of the difference between desired and measured value, and write it to an output file `error.txt`.

Example:

desiredsignal.txt	measuredsignal.txt	error.txt
100	99	1
110	111	1
120	112	8
120	113	7
120	115	5
120	116	4
120	118	2
130	127	3
130	128	2
130	131	1
140	134	6
140	145	5
140	140	0

You can download sample input files from the web page.

Take care that the two files `desiredsignal.txt` and `measuredsignal.txt` might have different length. The absolute value can be obtained by using the `abs()` function.

Exercise 4.2 (Maps)

Write a program that reads a text from a file and counts for each word the number of its occurrences in the text. You can download a sample input file `words.txt` from the web page.

Remark: For simplicity, you may assume that the input text does not contain any punctuation characters (like period, comma, semicolon, etc.), and is in lowercase only. The sample input file is in this format.

Exercise 4.3 (Maps)

Create a file `dict.txt` which contains a small English-German dictionary. The lines in `dict.txt` should have the following form:

```
morning Morgen
day Tag
hello Hallo
```

Write a program which first reads in the vocables from `dict.txt`. Then, the user shall be able to enter English words and the program prints the corresponding German words or a message if a word is not in the dictionary.

How can the program be modified such that translations in both directions are possible?

Exercise 4.4 (Maps and Vectors)

Download the file `grades.txt` from the web page. It contains the grades of different students of a class. Read in this data and print the following information for each student:

- his/her name;

- his/her list of grades; and

- his/her average grade (with a precision of one digit after the decimal point).

Example: The first line of output should look as follows:

```
Ann: 4, 2, 1, 1, 1 avg.: 1.80
```