**Working with Hashing**

The goal of this lab is to develop a program in Java that counts how many times each of the words in a file appears in the file (frequency of each word within the file). For the program a word is a sequence of alphabetic lower-case characters separated by a space. There is no limit in the length of the words or the input file.

For example suppose that the file contains the following:

in the beginning god created the heaven and the earth

and the earth was without form and void and darkness was upon the face of the deep and the spirit of god moved upon the face of the waters

and god said let there be light and there was light

and god saw the light that it was good and god divided the light from the darkness

and god called the light day and the darkness he called night and the evening and the morning were the first day

The output of the program after using as input this paragraph (it has 88 words) is as follows:

The paragraph has 88 words. The frequency of the letters is as follows:

the : 17

and : 13

god : 6

light : 5

was : 4

darkness : 3

of : 3

upon : 2

there : 2

face : 2

earth : 2

day : 2

called : 2

heaven : 1

he : 1

good : 1

from : 1

form : 1

first : 1

evening : 1

divided : 1

deep : 1

created : 1

beginning : 1

be : 1

without : 1

were : 1

waters : 1

void : 1

that : 1

spirit : 1

saw : 1

said : 1

night : 1

moved : 1

morning : 1

let : 1

it : 1

in : 1

**Requirements:**

* Create a class called Word that represents a word with its corresponding frequency
* Create a class called FileWordCounter where the logic of the program should be implemented. The program must ask for the location (name) of the input file.
* Use the Java API classes for hashing (HashMap, HashTable classes) to keep track of the frequencies.
* The output of the program should provide the words ordered by frequency (from the most frequent to the least frequent). Use Collections.sort to sort the words based on their frequency.
* The output should be printed in the screen and also be saved to a file. To create a file, utilize the same directory of the input file, and the name it with the name of the input file\_[date\_in\_millis]. For example, let’s say that the file is called propjkv.txt and it is in the C:\text\ folder. The output directory of the file is C:\text\ and the file name is propjkv.txt.109384848596.
* Test your programs on the three attached files. These files contained the same information, but they’re trimmed (less words). Try your program from the smaller to the greatest file.