

Assignment 9: File Analysis

CIS 210 – Fall 2016

Point: 100

Project Due: 11:59 PM on Thursday, 11/10/2016

Learning Objectives:

- ❖ To use Java APIs for text processing
- ❖ To read data from an input file
- ❖ To write data to an output file.

Problem Statement

You are required to write a program to read the content of an input file named *input.txt* and then write the following statistics into an output file named *output.txt*:

- The content of the input file
- The number of lines
- The number of words
 - Note: the character ' ' is one word. For example, the phrase “that’s” contains two words.
- The number of sentences
 - Every sentence ends with either a comma or semicolon.
- The number of punctuation characters
 - Punctuation is anything that neither an alphabet nor a digit
- The number of alphabet characters
- The number of digit characters
- The number of whitespace characters
- The number of words that start with uppercase letter
- The frequency of words (case insensitive). Please display and write the frequency of words as a list.
 - Frequency of a word shows the number of occurrences of a word in the input file. For example, this sentence “The sun is so nice in the morning” has the following frequency of words (case insensitive):
the: 2
sun: 1
is: 1
so: 1
in: 1
morning: 1

Input

This program requires that you read in data from an input file *input.txt*. You must ask the user to enter this file name. If the file doesn't exist, then display a warning message and quit the program.

Output

Your program should display the following on the console and write it to an output file named *output.txt*: (Please ask the user to enter an output file name)

- The content of the file.
- The number of words.
- The number of sentences.
- The number of alphabet characters.
- The number of punctuation characters.
- The number of digit characters.
- The number of whitespaces.
- The number of words that start with uppercase letter.
- The frequency of words.

Sample Output:

SIMPLE is an acronym for Sheer Idiot's Monopurpose Programming Language Environment. This language, developed at the Hanover College for Technological Misfits, was designed to make it impossible to write code with errors in it. The statements are, therefore, confined to BEGIN, END and STOP. No matter how you arrange the statements, you can't make a syntax error. Programs written in SIMPLE do nothing useful. Thus they achieve the results of programs written in other languages without the tedious, frustrating process of testing and debugging.

```
-----  
Total number of lines in the files : 8  
Total number of words in the files : 86  
Total number of sentences in the files : 6  
Total number of punctuation in the files : 15  
Total number of alphabets in the files : 448  
Total number of digits in the files : 0  
Total number of whitespaces in the files : 81  
Total number of uppercase in the files : 39  
-----
```

```
a: 1  
achieve: 1
```

acronym: 1
an: 1
and: 2
are: 1
arrange: 1
at: 1
begin: 1
can: 1
code: 1
college: 1
confined: 1
debugging: 1
designed: 1
developed: 1
do: 1
end: 1
environment: 1
error: 1
errors: 1
for: 2
frustrating: 1
hanover: 1
how: 1
idiot: 1
impossible: 1
in: 3
is: 2
it: 2
language: 2
languages: 1
make: 2
matter: 1
misfits: 1
monopurpose: 1
no: 1
not: 1
nothing: 1
of: 2
other: 1
process: 1
programming: 1
programs: 2
results: 1
sheer: 1
simple: 2
statements: 2

stop: 1
syntax: 1
technological: 1
tedious: 1
testing: 1
the: 5
therefore: 1
they: 1
this: 1
thus: 1
to: 3
useful: 1
was: 1
with: 1
without: 1
write: 1
written: 2
you: 2

Use of Methods, Parameters, Modularity, Design, etc.

Part of your grade on this and ALL future programming projects in this course will be determined by how well you use multiple functions and parameter passing appropriately and how well you design a modular and functionally cohesive program using the principles discussed in class. Large grade point penalties can be incurred for not setting up a modular, well designed program structure. This emphasizes good program structure, design, and fundamental software engineering principles.

Miscellaneous

Be sure to read and understand the sections in the Course Syllabus handout on general project requirements. Also be sure to study the style, documentation and formatting guidelines discussed in the *Programming Style Guidelines* handout and in the lecture.

What File To Turn In and How to Turn In Your Work using Blackboard

You must turn in your Java program source file. The file's name must follow this format:

`hw9_file_analysis.java`

Hence, file names will look something like "hw9_file_analysis.java"