2020 – 2021
FALL SEMESTER

CME 1203

INTRODUCTION TO COMPUTER ENGINEERING

ASSIGNMENT

E-BOOK ANALYSIS AND REPRESENTATION

DUE DATE: 13/01/2021 – 23.59

HOMEWORK DEFINITION

You are asked to design a Python program (in a single script file) to download e-books from (*Wikibooks or Wikisource*) and **save them to a text file**. The name of the book will be taken as user input. For this, you can use Web scraping libraries in Python to get requested books.

Assuming that you have downloaded the requested e-book from Wikibooks to a text file in your computer (should be in the same directory as your source code, do not use full file paths like "C:\Users\User\Desktop\…", just use file names)

After correctly saving the e-book to a text file, your program should be able to **read it and create word frequencies of that book**, meaning counting the number of times words has been used in that e-book. During this operation, you should also **remove stop words** (the, a, he, she, it, etc.), words that are meaningless by themselves but usually have the highest frequency of available words. You are required to use Python programming language and if you wish, you can use standard Python libraries. However, you are required to explain how you used it and what it actually does, in your source code comments**. Any tool that takes only inputs from you**, without any effort on your part to use it in Python, **will not be accepted**.

Another functionality you are expected to implement is the **comparison of two e-books and their word frequencies**. You will calculate word frequencies of two e-books and show the sum of frequencies of common words (words that appear on the both e-books) and show the frequencies of distinct words (words that only appear on only one of the e-books). The format you are required to follow with these requested outputs will be given later in this document.

Your program should ask the user, **how many word frequencies they wish to see** (in this document, as you can see, 20 words per list has been shown). You can take **20** as your default number if the user does not enter a specific number they wish to use.

Make sure that your program works on **command console**, rather than requiring an IDE to work (you can still use any IDE you wish to develop your assignment but you need to run your program on console after your development). We will evaluate your assignments by executing them on command console, as it has been done in our lab sessions. If your assignment cannot work by applying this method, we will not be able to evaluate them and you will get a **zero grade**.

You should **prepare a report** to explain your project development process. It will include structure of your project, algorithms you developed, code you write, use cases etc. Please **do not forget to add references you used** during development of your homework.

Please follow the rules below for correct assignment upload to the Sakai system. You should upload **a single Python file** containing your program and **a PDF file** for your report, named in the following formats:

```
<Student_Number>_<First_Name>_<Middle_Name>_<Last_Name>.py
<Student_Number>_<First_Name>_<Middle_Name>_<Last_Name>.pdf
```

Capitalize only the first letters of the names. Do not use Turkish characters. Do not zip the source file.

Cheating and plagiarism is strictly **prohibited**, and if it is detected, your assignment will be **graded 0**. Do not copy source code from the Internet, it will be treated as plagiarism. Late submission is allowed, however your grade will be reduced **20** points per day of late submission.

EXAMPLES OF REQUESTED OUTPUT

For this example, we will consider (Wikibooks contributors, 2020b) and (Wikibooks contributors, 2020a) e-books. To calculate their word frequencies, we will use (*Online Word Counter*, n.d.) website, an online service for text analysis. You can use this tool to compare your results and check their accuracy. Assume that when the user selected the e-book "Non-Programmer's Tutorial for Python 2.6", the console output is expected to be similar to the table below (try to make your console output as orderly as possible, learn about advanced python console output and string manipulation for this purpose):

```
BOOK 1: Non-Programmer's Tutorial for Python 2.6
NO WORD     FREQ_1
 1 print    520
 2 number   268
 3 program  179
 4 python   158
 5 +        151
 6 input    141
 7 list     137
 8 function 131
 9 menu     100
10 true      96
11 type      92
12 item      91
13 string    87
14 license   82
15 numbers   82
16 document  75
17 file      75
18 text      72
19 return    68
20 false     67
```

If the user selected only one e-book, this will be the end of the program. However, if the user selected two e-books, the output should be similar to the table given below:

```
BOOK 1: Non-Programmer's Tutorial for Python 2.6
BOOK 2: Non-Programmer's Tutorial for Python 3
COMMON WORDS
NO WORD     FREQ_1 FREQ_2 FREQ_SUM
 1 print    520    529    1049
 2 number   268    288     556
 3 program  179    177     356
 4 python   158    198     356
 5 list     137    157     294
 6 +        151    137     289
 7 input    141    132     273
 8 function 131    123     254
 9 menu     100    109     209
10 true      96     99     195
11 numbers   82    111     193
12 type      92     95     187
13 item      91     89     181
14 file      75     96     172
15 string    87     83     170
16 choice    66     75     142
17 false     67     69     136
18 text      72     55     127
19 return    68     56     124
20 var       62     62     124
```

BOOK 1: Non-Programmer's Tutorial for Python 2.6
DISTINCT WORDS

NO	WORD	FREQ_1
1	document	75
2	raw	66
3	sections	31
4	title	28
5	invariant	23
6	modified	20
7	texts	20
8	cover	16
9	include	16
10	entitled	15
11	distribute	13
12	preserve	13
13	publisher	12
14	documents	11
15	transparent	11
16	finally	9
17	gnu	9
18	history	9
19	mmc	9
20	published	9

BOOK 2: Non-Programmer's Tutorial for Python 3
DISTINCT WORDS

NO	WORD	FREQ_2
1	path	11
2	python3	11
3	subprocess	10
4	click	8
5	environment	8
6	pip	7
7	directory	6
8	wt	6
9	arithmetic	5
10	imported	5
11	rt	5
12	select	5
13	spam	5
14	started	5
15	545-4464	4
16	\tnumber	4
17	bigger	4
18	button	4
19	closing	4
20	https	4

ASSIGNMENT GRADING TABLE

REQUIREMENT	**GRADE**
Correct upload file format.	5
Writing understandable and detailed comments.	5
Using good and understandable variable, function and etc. names.	5
Downloading e-book correctly.	10
Reading a text file correctly.	5
Removing stop words.	5
Writing calculated frequencies on console in understandable and orderly format.	10
Calculating word frequencies for one file correctly.	15
Calculating word frequencies for common words of two files correctly.	15
Calculating word frequencies for distinct words of two files correctly.	15
Report (PDF)	10
TOTAL	**100**
Late submission per day.	**-20**
Cheating or any other form of plagiarism.	**-∞**

REFERENCES

Online Word Counter. (n.d.). CountWordsFree. Retrieved December 22, 2020, from https://countwordsfree.com

Wikibooks. (n.d.). Retrieved December 22, 2020, from https://en.wikibooks.org/wiki/Main_Page

Wikibooks contributors. (2020a). *Non-Programmer's Tutorial for Python 3—Wikibooks, open books for an open world*. Wikibooks, The Free Textbook Project. https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3

Wikibooks contributors. (2020b). *Non-Programmer's Tutorial for Python 2.6—Wikibooks, open books for an open world*. Wikibooks, The Free Textbook Project. https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_2.6