How to use SearchEngine.java

- SearchEngine.java is a a program that runs on the command line.

- In order to run the the program on the command line you must enter certain arguments.

- The command line should look as so :

- java SearchEngine -CorpusDir <PathOfDir> -InvertedIndex <NameOfIndexFile> -StopList <NameOfStopListFile> -Queries <QueryFile> -Results <ResultsFile> -SearchWord <Word> -SearchDoc <DocName> -WriteDecision <yes/no> -Stem <yes/no> -Snippet <integer value> -Output <gui/file/both>

- java SearchEngine -h

- By saying "java SearchEngine -h" this will print the manual for the SearchEngine's command line as well.

- java SearchEngine must always be said first, everything else can be said in any other order but the the name you want to assign to the variable must always follow the variable name, for example, -CorpusDir must always be followed by <PathOfDir> or the program will not compile correctly.

- The variable names such as -CorpusDir and -InvertedIndex must follow the exact syntax, must have uppercases where uppercase is present and the same for lowercase, no space between "-" and the variable, and they must have a space between the variable name and the value you want to assign to it, -CorpusDir<PathOfDir> is not valid.

- The words in "<>" are the names you will provide, but you will provide it without the "<>", for example "<PathOfDir>" must be provided as "PathOfDir".

- -CorpusDir <PathOfDir> is the path of the the directory which contains the files you want to read.

- -InvertedIndex <NameOfIndexFile> is the name of the file that you want to print the inverted index to, be sure to include the file type, .txt, .html and so on.

- -StopList <NameOfStopListFile> is the name of the stop list file that contains the stop words you would like to skip over when reading the files.

- -Queries <QueryFile> is the name of the file that contains the queries you want, it must be formatted properly, if you want to query the word hello the query file must be formatted as : Query hello , if you want to query something else in the same file you must use a new line. The syntax must be followed, "Query" must have an uppercase Q and the query word must be in all lowercase because the program makes "Hello" or "HELLO" into "hello" since we don't want multiple variations of the same word.

- -Results <ResultsFile> is the name of the file that you want to output the queries to, it will output in how many documents the query appears, in which specific document, how many times it appears in that specific document and in what position the query appears for the specific document.

- -SearchWord <Word> this is a word that you would like to query and it will return all the information associated with that query in the ResultsFile or to the console, the user decides this in -WriteDecision <yes/no>. If the word does not exist or if the word entered is a stopword, this will also be printed.

- -SearchDoc <DocName> this is a specific document that you would like to see all the information of, make sure to include the file type, .html, so if the document is named "hello" and it's of file type html, it must be entered as "hello.html". The information that will be printed includes how many unique words appear in the document, every specific word that appears in the document, for each specific word it will tell you how many times that word appears in the document and in what positions that specific word appears in. All this information will be printed either to the ResultsFile or to the console, the user decides this in -WriteDecision <yes/no>.

- -WriteDecision <yes/no> is a decision the user makes for where they want to print SearchWord and SearchDoc, if the user wants to print SearchWord and SearchDoc to the ResultsFile then input "yes" if the user wants to print SearchWord and SearchDoc to the console then input "no" for this variable. You cannot decide to print only SearchWord to the console and SearchDoc to the ResultsFile, they will both print to the same place.

- -Stem <yes/no> is a decision the user makes for whether they want to stem the words from the query file, if the user decides yes then the queries from the query file will be stemmed and will be printed to the results file. The printing of the stemmed words will come after the printing of the original query words so to see the results of the stemmed queries you must go below the original queries.

- -Snippet<integer value> is the amount of words before and after the query that you want to see, so for example, -Snippet 8 would print 8 words before the query, the query word and then 8 words after the query. If the it prints less than 8 words before or after this means that some of the words before and after the query we're stopwords so we never saved their position since it isn't a valid word to query.

- -Output <gui/file/both> is where you want the results of the query file to be printed, if you choose gui it will print out to a gui only, if you choose file, it will print out to a file only and if you choose both, it will print out to both the gui and the file. The file name must be provided in the -Results <ResultsFile> flag, here is where the printed file will be output to if "file" or "both" is chosen.

- Once all this information is provided then the program will run properly and will output all the information of all the documents.

- The output includes the words in every document and in which document it appears in, in which spot in that specific document, how many times in that specific document and it also contains a snippet.

- In the query file some words that are input are important key terms in relation to the original queries. For example, the query "Who is the president", in this query the word "president" is extremely important so because of it's importance the precision and recall of this word is printed in addition to the other information.

- Below is a picture of how the GUI will look after it is printed to, it also gives a great example of how the output looks.

