ACS-1903-**050** Assignment #2 Due Sunday Nov 10, 2019 at 11:59 pm

Copy the file, HP.txt[1], on the course web page to Readme.txt in your BlueJ project.
Develop and test a Java program, AnalyzeText, that reads Readme.txt and reports the following:

a) The 3 most common words **that are not** in this stop-list:
```
a,am,an,and,any,are,as,a,be,by,he,her,hers,him,his,i,if,in,into,is,it,its,
me,my,no,nor,not,of,on,or,she,than,that,the,their,them,then,there,these,
they,to,too,us,was,we,were,what,when,where,which,while,who,whom,why,you
```
An example of output:
```
Word    Frequency
wand    21
potion  10
wizard  9
```
b) The 3 most common names **that are** in this list of names:
```
Harry, Dumbledore, Voldemort, Snape, Sirius, Hermione, Ron, Draco, Hagrid,
Neville, Dobby, Moody, Lupin, Bellatrix, McGonagall, Grindelwald, Tina
```
An example of output:
```
Name    Frequency
Harry   15
Snape   8
Dobby   5
```

Regarding Part A:

Consider the following approach to Part A.
For Part A you determine the vocabulary of the text, and for each word you determine the frequency. On paper the vocabulary and frequencies comprises two parallel lists, one for words and one for frequency.

word	frequency
star	5
potion	10
wizard	9
wand	21
etc	

An algorithm in **pseudocode to create the two lists**:

```
For each token in Readme.txt:
        get the token (as lowercase) with punctuation removed
        if the token is not in the stop list
           if the token is in the word list
                increase its frequency count
           otherwise
                add the token to the word list
                set its frequency to 1.
//
// determine 3 most frequent words
Repeat the following 3 times:
        Find the largest frequency and then:
```

[1] *Rowling, J. K. Harry Potter And the Deathly Hallows. New York, NY : Arthur A. Levine Books, 2007.*
http://hpread.scholastic.com/HP_Book7_Chapter_Excerpt.pdf

report the corresponding word
set that word's frequency to -1

When you implement Part A you must develop and use these methods:

Return type	Method name and parameters	Purpose
`String`	`getToken( Scanner s )`	Finds the next token using the `scanner` and returns the token in lowercase with punctuation characters removed.
`int`	`find( ArrayList<String> word, String token )`	Returns the index of `token` in `word`, or -1 if not present.
`void`	`incrementFrequency( ArrayList<Integer> freq, int i )`	Increments the `i`th entry in `freq` by 1
`void`	`addNewWord( ArrayList<String> word, ArrayList<Integer> freq, String token )`	Adds the `token` to `word`, and adds a corresponding entry to `freq` with its value set to 1.

With the above methods the code to process a single token is

```
String token = getToken(s);
if (! stopList.contains(token)){
int i = find(word, token);
if (i>=0) {
    incrementFrequency(freq, i);
}
else {
    addNewWord(word, freq, token);
}
```

Regarding Part B: Use the above methods as much as possible

All classes **must** have comments at the beginning containing your name and student number.

Submit the file `AnalyzeText.java` to the email corresponding **to your lab section** with a **Subject line** Assignment 1
E.g. if you are registered in lab ACS-1903L-070 then send to 1903L-070@acs.uwinnipeg.ca