COP 3003 Object-Oriented Programming, Fall, 2016

Lab 5
Due: 11:59pm November 18, 2016
Important Notes:

1. Your programming format must be consistent with the conventions introduced in code_conventions.ppt. Failure to do that may cost you up 10% of your grade.
2. Please remove or do not any package statement in your Java programs. This can help TAs grade your program.
3. Do not write your name anywhere in your program. Your assignment will NOT be grade if this requirement is not met.
4. When submitting, zip your source code and submit it on Canvas. Please also name your zip file with the last four digits of your UIN.
5.

Assignments:

In this lab assignment, you are given a text file, basketball.txt. You are supposed to write a program to examine the words in this file and find how many times each word appears in the text. At last, display the FOUR most frequent words and their frequencies (number of appearances).

Note you are supposed to use the existing JDK classes and methods for counting the frequencies and finding the most frequent words.

In the class, you have learned how to use Map to find frequencies. In this assignment, you are required to learn and use Collection.frequency(…) to find frequencies.

First of all, please declare the following instance variables and private class in class FileStats:

```
private Scanner input=null;
private ArrayList <String> wordList=new ArrayList<String>();
private HashSet <String> wordSet=new HashSet<String>();
private ArrayList <Entry<String>> entryList=new ArrayList<Entry<String>>();

private class Entry <T> implements Comparable<Entry<T>>{
        public T word;
        public int frequency;
        public Entry(T word, int f){
                this.word=word;
                frequency=f;
        }
        public int compareTo(Entry<T> e){
                /* insert your code here */
        }
}
```

wordList is used to store all the words in the file and is expected to have duplicates. wordSet only stores distinct words without duplicates. We will come back and see what entryList means.

Now let's find how to get individual words from the file. You have learned class Scanner. A line can be read from a file by the following code:

```
try{
        input=new Scanner(new File(path));
}catch(FileNotFoundException e){
        System.out.println("Error openning file..");
        System.exit(1);
}

try{
        while((line=input.nextLine())!=null){
                /* insert your code here */
        }
}catch(NoSuchElementException e){
        // no more lines in the file
        // no handler is necessary
}
```

Now the question becomes how to get individual words from a line. In this lab, please use class StringTokenizer. This class enable you to break a string into tokens. The default delimiters are space, tab, new line, carriage return, and form feed. The following code shows the tokenization:

```
 StringTokenizer st = new StringTokenizer("this is a test");
while (st.hasMoreTokens()) {
    System.out.println(st.nextToken());
}
```

It displays the following:

```
this
is
a
test
```

We can have similar code for tokenizing a line for words. Note you do need to consider two situations:

- Some words are followed by punctuation. Hint: consider using String.subString(…)
- Same word can begin with either capital or lower case letter. In this assignment, your program is not case-sensitive. Look at String.toLowerCase() method.

For each word from the text, add it to both wordSet and wordlist. Then wordList will have all the words in the file and wordSet only has the distinct words.

Then, we are ready to find the frequency of each word in **wordSet**. You can do this by calling "Collections.frequency(**wordList**, word)" that can find, in wordSet, the frequency of each word in wordSet.

At last, let’s discuss class Entry and variable entryList. This class has two instance variables: word and frequency. Say you have the following entries in a list:

word	frequency
this	9
basketball	8
is	10
and	12
points	3

We can call Collections.sort(…) to sort the list to the following:

word	frequency
and	12
is	10
this	9
basketball	8
points	3

What you need to do to sort is to define the “compareTo” method in class Entry.

Now we can easily find the first four elements that are the most frequent words.

```
for(int i=0;i<4;i++){
        System.out.println(entryList.get(i).s+" appears "+
                entryList.get(i).frequency+" time(s).");
}
```

At last, let’s see where to put the given file. In Eclipse, put the given file in the root folder of the project. If you use notepad + JDK commands, put the file where the source file is.

The correct result is

```
the appears 10 time(s).
basketball appears 8 time(s).
and appears 6 time(s).
is appears 6 time(s).
```

Grading Policies:

Correctly read each word:	30%
Correctly build entryList:	30%
Successfully find 4 most frequent words:	30%
Correct output:	10%