Lab 10: Checking English Words

In this lab you will use a TreeSet object to store a list of English words. The class you turn in should be called WordChecker.

A TreeSet is one of Java's built-in classes. You can do an internet search for ”java treeset” to get its complete documentation. Since it is a set, it contains many generic objects, with no duplicates. Thus, you are allowed to put objects into the TreeSet, remove them, or query it to see if it contains a particular object. It is called a TreeSet because it stores the objects in a binary search tree for efficiency. However, this will all be done behind the scenes—you won't need to program your own tree.

The list will be an exhaustive list of valid English words, which is contained in the file english.lex. Your program will load this file, store all the words into the TreeSet, and then inform the user how many words there are. Then it will enter a loop, each iteration prompting the user for a word. If the word was present in the original list, it will output "<word>" is a valid word. where <word> is the word in question. Otherwise, it will output "<word>" is NOT a valid word. If the user hits enter without entering a word, the program will quit, like this:

```
Welcome to the Word Checker!
Loading file "english.lex", which contains 62970 words.
Please enter a word, or hit enter to quit:
> hello
"hello" is a valid word.
> giraffe
"giraffe" is a valid word.
> lskdfj
"lskdfj" is NOT a valid word.
>
Goodbye!
```

Some hints:

- You can use a Scanner to input the words from the .lex file. To do this, you must make a new File object, using the file's name as the sole argument to its constructor. If you then create a constructor using the File object as its constructor argument, the Scanner will read the list of words just like you had typed them in. Note however that making a new File object must be contained within a try-catch block, that checks for an IOException.
- Try to use no class variables (aside from constants) if you can. It is much better for functions to share information only through arguments and returned values.
- Sketch out ahead of time how you might want to divide the task into relatively independent functions. For example, a createTreeSet() function that takes an array of Strings and returns the TreeSet that holds them all might be a useful function. So

might a `queryUser()` function that does all the interaction with the user.

- A Windows computer might automatically change `english.lex` to `english.lex.txt`. This is incorrect, and you will not get full credit if your program does not use `english.lex`. If your computer helpfully makes this change, it is your job to helpfully change it back.