COS126 Symbol Table Activity - 4.4.36 (Booksite Creative Ex 4.4.5)

- Here is a subset of the API for the ST class you will use.

```
public class ST<Key, Value>         // Note: Key must be Comparable
------------------------------------------------------------------------
             ST<Key, Value>()       // create a symbol table
       void  put(Key key, Value v)  // put key-value pair into the table
      Value  get(Key key)           // return value paired with key
                                    // or null if no such value
    boolean  contains(Key key)      // is there a value paired with key?

NOTE: Allows iteration with enhanced for loops:
for (Key key : st) { . . . }       // executes body once for each key
```

- Here is the API for the FrequencyTable class you will write.

  A FrequencyTable represents a table that tracks the number of repeated occurrences of items in a list of Strings.

```
public class FrequencyTable
------------------------------------------------------------------------
             FrequencyTable()     // new table
       void  click(String word)   // add one to frequency of this word
        int  count(String word)   // what is frequency of this word?
       void  show()               // print all words and frequencies
       void  main(String[] args)  // build and print frequency table
                                  // of words on standard input
```

  For Example: If you run FrequencyTable with the following input:

```
duck duck goose
```

  you should see the following output:

```
 2 duck
 1 goose
```

- Recommended Book Exercise: ST client program 4.4.23

- Complete the code for the FrequencyTable class below.

```
// Dependencies: ST.java (available on precepts page), StdIn, StdOut
public class FrequencyTable {
    // maintain counts of all words seen so far
    // the key is the word and the value is the count
    private ST<___________ > st = _____________________________ ;

    // remark: We have no explicit constructor code!  But Java lets every class
    // have a no-argument constructor by default.  It only runs the line of
    // code above (instance variable declaration and initialization together).

    // add one to the frequency count for this word
    public void click(String word) {
        int count = count(word);
        st.put(word, count + 1);
    }

    // what is the frequency of this word?
    public int count(String word) {
        if (!st.___________(______)) return 0;  // if word is not in ST
        else return ______._________(______);   // get word’s count
    }

    // print out all words and frequencies (frequencies first)
    public void show() {
        // foreach loop. goes through all keys in alphabetical order
        for (String word : st) {
            // print out frequency and word, separated by a space
            StdOut.println(__________(____) + " " + _____________);
        }
    }

    // method used by client to count all words in StdIn
    public static void main(String[] args) {

        // build frequency table from words on standard input
        FrequencyTable freq = new FrequencyTable();
        while (!StdIn.isEmpty()) {
            String word = StdIn.readString();
            freq.__________(word);
        }

        // print frequency table to standard output
        freq.show();
    }
}
```