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

- Recommended Book Exercises:

  BST mechanics - 4.4.8, 4.4.9

  ST client program 4.4.23

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

```
public class ST<Key extends Comparable<Key>, Value>
-----------------------------------------------------------------------
             ST()                    // 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?
```

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

```
public class FrequencyTable
-----------------------------------------------------------------------
             FrequencyTable()        // do-nothing constructor
       void  click(String key)       // add one to the count for the key
        int  count(String key)       // number of times key appears
       void  show()                  // print each key preceded by count
       void  main(String[] args)     // build and print frequency table
                                     // of words on standard input
```

- Complete the code for the FrequencyTable class below.

```
 1:/*************************************************************************
 2: *  Compilation:  javac FrequencyTable.java
 3: *  Dependencies: StdOut.java StdIn.java ST.java
 4: *  Execution:    java FrequencyTable < words.txt
 5: *  Data file:    http://introcs.cs.princeton.edu/java/44st/mobydick.txt
 6: *
 7: *  Read in a list of words from standard input and print out
 8: *  each word and the number of times it appears.
 9: *
10: *  % java FrequencyTable < mobydick.txt | more
11: *  4583 a
12: *  2 aback
13: *  2 abaft
14: *  3 abandon
15: *  7 abandoned
16: *  1 abandonedly
17: *  2 abandonment
18: * . . .
19: *
20: *************************************************************************/
```

```
32:public class FrequencyTable {
33:    private ST<String, Integer> st = new ST<__________, ___________>();
34:
35:    // add one to the count for the key
36:    public void click(String key) {
37:        int count = count(key);
38:        st.put(key, count + 1);
39:    }
40:
41:    // return the number of times the key appears
42:    public int count(String key) {

43:        if (!st.___________(______)) return 0;  // if key is not in ST

44:        else return ______._________(______);   // get key's count
45:    }
46:
47:    // print each key preceded by its count to standard output
48:    public void show() {
49:        for (String key : st) {   // iterate for each key in st

50:            StdOut.println(__________(key) + " " + ____________);
51:        }
52:    }
53:
54:    public static void main(String[] args) {
55:
56:        // build frequency table from words on standard input
57:        FrequencyTable freq = new FrequencyTable();
58:        while (!StdIn.isEmpty()) {
59:            String key = StdIn.readString();

60:            freq.__________(key); // call method to increment key's count
61:        }
62:
63:        // print frequency table to standard output
64:        _________________();
65:
66:    }
67:
68:}
```