

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()                             // 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.

```

1:// Dependencies: ST.java (available on precepts page), StdIn, StdOut
2:public class FrequencyTable {
3:    // maintain counts of all words seen so far
4:    // the key is the word and the value is the count
5:    private ST<_____ > st = _____ ;
6:
7:    // remark: We have no explicit constructor code! But Java lets every class
8:    // have a no-argument constructor by default. It only runs the line of
9:    // code above (instance variable declaration and initialization together).
10:
11:    // add one to the frequency count for this word
12:    public void click(String word) {
13:        int count = count(word);
14:        st.put(word, count + 1);
15:    }
16:
17:    // what is the frequency of this word?
18:    public int count(String word) {
19:        if (!st._____(_____)) return 0; // if word is not in ST
20:        else return _____(_____); // get word's count
21:    }
22:
23:    // print out all words and frequencies (frequencies first)
24:    public void show() {
25:        // foreach loop. goes through all keys in alphabetical order
26:        for (String word : st) {
27:            // print out frequency and word, separated by a space
28:            StdOut.println(_____(____) + " " + _____);
29:        }
30:    }
31:
32:    // method used by client to count all words in StdIn
33:    public static void main(String[] args) {
34:
35:        // build frequency table from words on standard input
36:        FrequencyTable freq = new FrequencyTable();
37:        while (!StdIn.isEmpty()) {
38:            String word = StdIn.readString();
39:            freq._____(word);
40:        }
41:
42:        // print frequency table to standard output
43:        freq.show();
44:    }
45:}

```