121 Discussion #8 3/22/10
OWL #7 now working correctly;
Program 5 – letter frequencies, due tonight. Tomorrow: recursion
– trapdoor from ch 8. Covered before break – big programs: multi-
constructors, signatures, string tokenizer, string equality.

What does this program do??

```java
import java.util.*;

public class Lengths{// 2 CRs ends it
    public static void main(String[] args){
        Scanner scan = new Scanner(System.in);
        int pos = 0;
        String t = " ";
        while(t.length() > 0){
            t = scan.nextLine();
            System.out.println(pos + "-"+ t.length());
            pos++;
        }
    }
}

Tokenizer. Lines **, ***, below, are alternatives. What’s the difference? How does each handle
this phrase: “the quick, near-brown fox: it jumped!!”

import java.util.StringTokenizer;
import java.util.Scanner;
public class TokenizerTest{
    public static void main(String[] args){
        StringTokenizer str;
        Scanner scan = new Scanner(System.in);
        System.out.println("enter a line of text");
        String s = scan.nextLine();
        str = new StringTokenizer(s); // ** or
        str = new StringTokenizer(s," .,!?;:"); // ***
        while (str.hasMoreTokens()){
            System.out.println(str.nextToken());
        }
    }
}
```

1. Now: edit TokenizerTest so that it expects, and reads in a single string of A’s and B’s only, then reports the number of blocks of B’s it sees. Example: “AABBBABBBAAABAB” -> 4

2. Edit TokenizerTest so that it reads in a single string, and then prints the number of words in the string that are strictly longer than 5 letters.

3. A sentence is a string that ends with an !,?, or . Rework Lengths (above) so that it reads lines from the keyboard, then prints each sentence (with punctuation missing) on a separate line. (one way: first turn the input into one long string.)