```
import java.util.*;                                                                                          // need this for Scanner
class WordCount

   {

      /* PROMPT USER FOR STRINGS AND DISPLAY WORD COUNT, UNTIL USER ENTERS SENTINEL */

      public static void main (String [ ]  args)

         {

            final String SENTINEL = “”;                                                       // sentinel string that terminates program

            Scanner keyboard = new Scanner (System.in);                          // Scanner object wrapped around the keyboard, for user input

            // prompt user and get first input string

            System.out.print  (“\nEnter a string of characters (” + ( SENTINEL.isEmpty( ) ? “enter” : (“\”” + SENTINEL + “\””) ) + “ to quit):
                               ”);

            String input = keyboard.nextLine ( );


            // If input is not SENTINEL, display word count, and then prompt and input again and display again, repeatedly

            while ( !input.equals (SENTINEL) )                    // SENTINEL string terminates loop, which processes each input string in turn

               {

                  StringTokenizer word = new StringTokenizer (input);                  // break up input into sequence of its constituent words

                  String [ ] wordCollection = new String [words.countTokens ( )];               // stores words as they are encountered
```

```
int numberOfWords = 0;                                                  // number of distinct words encountered so far

int [ ] wordCount = new int [words.countTokens ( )];          // wordCount[i] is number of occurrences – 1 of wordCollection[i] in line

int i;   // general-purpose index used to step through the entries of wordCollection in the search for match with word, and to display words and counts

while ( words.hasMoreTokens ( ) )                               // keep going while there are still words left in the line to process

  {

    String word = words.nextToken ( );                                      // word is current word of line we are processing

    wordCollection[numberOfWords] = word;    //  put word at the end to guarantee loop termination, since we’re guaranteed to find a match

    for  (i = 0; !wordCollection[i].equals (word); i++) ;            // search wordCollection to determine if word is already stored in it

    if  (i = = numberOfWords)  numberOfWords++;          // word wasn’t there previously; now it’s in because we inserted 2 lines above

    else wordCount[i]++;                    // word was found there, so increment counter that tracks number of occurrences of word in line

  }                                                                          //  end of while that processes all words in line

for (i = 0; i < numberOfWords; i++)                                  // display contents of wordCollection and wordCount

  System.out.println (wordCollection[i] + “        ” + (wordCount[i] + 1) + (i < numberOfWords – 1 ? “” : “\n”) );    // “\n” at end


// prompt user and get next input string

System.out.print  (“\nEnter a string of characters (” + ( SENTINEL.isEmpty( ) ? “enter” : (“\”” + SENTINEL + “\””) ) + “ to
                       quit):  ”);

input = keyboard.nextLine ( );
```

```
        }                                                      // end while
    }                                                          // end main
}                                                              // end class WordCount
```