**Lab 27 – Word Scramble**

Open BlueJ, and create a new BlueJ project titled **Lab27-WordScramble** in your CS\LABS folder.

Create a new class called **PracticeProblems**:

In this lab, we’ll be working with a new data type: char. A char is a container than holds a single character (letter, digit, punctuation, etc.). A char is different from a String (even a String that only contains one letter!). A char literal is indicated using a pair of single quotes, like this: 'a'. The charAt() method will be particularly useful in this lab.

**Before each problem, insert a COMMENT with the problem number.**

1. Complete the method: isLetterA(char letter), that returns true if the supplied parameter is the character ‘a’.
2. Complete the method: hasMultipleAs(String s), that returns true if the supplied parameter contains at least two ‘a’ characters. This method should utilize (call) the method you wrote in practice problem 1.
3. Complete the method: isVowel(char letter), that returns true if the supplied parameter is a vowel.
4. Complete the method: numVowels(String s), that returns the number of vowels in the String parameter. This method should utilize (call) the method you wrote in practice problem 3.
5. Complete the method: evenChars(String s, char a, char b), that returns true if there are an equal number of char aand char b in the String parameter.
6. (Riddle) Four jolly folks sat down to play, they played all night until the break of day. They played for cash and not for fun, with a separate score for every one. When it came time to square accounts, they all made quite fair amounts. Somehow, no one lost and all gained. How is this possible?
7. Complete the method: hasRepeat(String s, char a), that returns true if a occurs twice in a row in the String parameter.
8. Complete the method: capChar(char a), that returns a capitalized version of the character received. Remember the toUpperCase() method? It only works on Strings, but you can still use it to your advantage here! You just have to be clever. In fact, the entire body of the method can be written as a single line of code! (Hint: concatenating a char and a String gives a String.)
9. Complete the method: capVowels(String s), that returns a new String where all the vowels (and only the vowels) in s are capitalized. **You MUST call 2 of your previous methods inside of this method.** (Hint: build a new string rather than altering the original.)

**Word Scramble**

*“Adcncriog to a reescarh sduty at Cimbrdgae Uivrnstiey, it dnose’t mtaetr in what oedrr the letters in a word are, the only itnpraomt tnhig is taht the first and last lteetr be in the rhigt pclae. The rest can be a toatl mses and you can slitl raed it whtiout a perlbom. This is becsaue the haumn mind does not read every ltteer by itlsef, but the wrod as a whloe.”*

Create a new class called WordScramble.

Complete the method: public static void scramble(String line), that takes a String parameter line and scrambles the interior letters of **each word** and prints a ‘scrambled’ version of line to the console (screen). **The parameter** line **can be any number of words!** Requirements:

* The first and last letters of each word are to remain in place. Only interior letters will be scrambled.
* Trailing punctuation (such as periods, commas, question marks, etc) should be retained in its place and not counted as the last letter in the word. Interior punctuation (like apostrophes) may be scrambled.
* Letter case should not be altered.

Examples:

understand ustdnread

what what

don’t dn’ot

hello hlleo

however, hveewor,

Never! Neevr!

Your program (class) should contain a *main()\** method. In the *main()* method, you should get a String of text from the keyboard, and call the *wordScramble()* method on that String.

\*The *main()* (public static void main(String[] args)) method is a special method in Java and is the method Java will look for when running a program. Note that in BlueJ, you can still run programs without a *main()* method, but that is not the case for most IDEs.

**There are many ways to solve this! Code it in any way you’d like, but if you need help, one solution is described on the next page.**

**Looking for help on WordScramble, huh?**

Some helpful methods that you might not know yet are **split()**, **Character.isLetter(), and toCharArray()**. Look these methods up on your own! You’ll be glad you did!

**split()** can help you break the message into its words. **isLetter()** can help you determine if you are looking at punctuation. toCharArray() can help you break each individual word into an array of characters (just like in Hangman).

But for each word, how could you scramble the interior letters?

['s','o','m','e','t','h','i','n','g'] <<< original word, in a char array

0 1 2 3 4 5 6 7 8

Run a series of letter swaps where **each position 1 through 7** gets the opportunity to swap with another (randomly chosen) letter from positions 1 through 7.

0 1 2 3 4 5 6 7 8

['s','e','m',**'o'**,'t','h','i','n','g'] <<< 1st swap (Pos 1 randomly chose to swap with Pos 3)

['s','e',**'h'**,'o','t',**'m'**,'i','n','g'] <<< 2nd swap (Pos 2 randomly chose to swap with Pos 5)

['s','e','h',**'m'**,'t',**'o'**,'i','n','g'] <<< 3rd swap (Pos 3 randomly chose to swap with Pos 5)

['s','e','h','m',**'t'**,'o','i','n','g'] <<< 4th swap (Pos 4 randomly chose to swap with Pos 4)

['s','e',**'o'**,'m','t',**'h'**,'i','n','g'] <<< 5th swap (Pos 5 randomly chose to swap with Pos 2)

['s','e','o',**'i'**,'t','h',**'m'**,'n','g'] <<< 6th swap (Pos 6 randomly chose to swap with Pos 3)

['s','e','o','i','t',**'n'**,'m',**'h'**,'g'] <<< 7th swap (Pos 7 randomly chose to swap with Pos 5)

Now every “swappable” letter has had the opportunity to move around. Your word has changed from “something” to “seoithmhg”. Notice that positions 0 and 8 never changed. You’ll have to make sure your loop and random number chooser are set up correctly so that these positions are never considered for swapping.

How do you swap 2 letters? Well, it takes only 3 lines of code, but you need an extra variable to use as a temporary placeholder.

Psuedocode for swapping two things:

Copy the value of thing #1 into a temporary variable.

Copy the value of thing #2 into thing #1.

Copy the value of the temporary variable into thing #2.

\*/