**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.”

You will write a program that reads in a file, scrambles the letters, and saves to an output file.

1. Create a class called WordScramble. It will have a main method.
2. In main, read in the input file called WordScrambleInput.txt by creating a loop to read word by word.
3. Write a method mixWord that receives a string as a single word and scrambles the interior letters and returns it.
   1. First and last letters are to remain in place. Interior letters will be scrambled.
   2. Leading and trailing punctuation will be retained in their places and not counted as the first or last letters in the word. Interior punctuation may be scrambled.
   3. Letter case should not be altered.

|  |
| --- |
| WordScramble |
|  |
| public static void main (String args[])  private static String mixWord(String word) |

understand ustdnread

what what

don’t dn’ot

“hello” “hlleo”

however, hveewor,

“Bygones!,” “Bnegyos!,”

1. In main, create an output file of your text with scrambled words.

**Rubric**

|  |  |  |
| --- | --- | --- |
|  |  | **Points** |
| main method | Method header | 5 |
| File input | Use of Scanner to read in a file | 20 |
| Process all words | Call to mixWord method for each word and output scrambled words | 25 |
| mixWord method | Method header and return statement | 25 |
| Identify punctuation | Move indexes to interior letters | 10 |
| Scrambles interior letters | Shuffle letters using random indexes | 15 |
|  | **Total** | **100** |