|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| LEARNING PROFILE FOR Chapter04Exercise01.printCapitalized(String) | | | | | |
| *Name* | *:* | *Tyler Lucas* | *Submission Date* | *:* | *N/A* |
| *Student ID* | *:* | *3305203* | *Date Created* | *:* | *2017-05-19* |

# Problem Statement

To "capitalize" a string means to change the first letter of each word in the string to upper case (if it is not already upper case). For example, a capitalized version of "Now is the time to act!" is "Now Is The Time To Act!". Write a subroutine named printCapitalized that will print a capitalized version of a string to standard output. The string to be printed should be a parameter to the subroutine. Test your subroutine with a main() routine that gets a line of input from the user and applies the subroutine to it.

Note that a letter is the first letter of a word if it is not immediately preceded in the string by another letter. Recall from Exercise 3.4 that there is a standard boolean-valued function Character.isLetter(char) that can be used to test whether its parameter is a letter. There is another standard char-valued function, Character.toUpperCase(char), that returns a capitalized version of the single character passed to it as a parameter. That is, if the parameter is a letter, it returns the upper-case version. If the parameter is not a letter, it just returns a copy of the parameter.

# Description of the Code

Gets string input from user and echoes it with each word capitalized.

# Errors and Warnings

No errors.

# Sample Input and Output

## [Version 1.0]

Enter phrase to be capitalized: test Test TEST tEST 0test0Test0TEST0tEST0 test Test TEST tEST (last set of "tests" have non-breaking spaces between them)

Capitalized version: Test Test TEST TEST 0Test0Test0TEST0TEST0 Test Test TEST TEST (Last Set Of "Tests" Have Non-Breaking Spaces Between Them)

# Discussion

No errors. Everything seems to work.

In the loop checking each character and the one before it to see if it’s the first letter of a new word, I opted to store a copy of the characters and the Boolean values of whether they were letters. This halved the number of times I had to look up those values.

I used the StringBuilder class, one I haven’t had much experience with, as it allows replacing individual characters in its string-like array, where String does not (java String objects are ‘immutable’, meaning they can’t be changed).