Characters & Arrays

- A String object cannot be manipulated as a set of characters
- However,
  - the string stored in a String object **can** be converted to a `char` array and,
  - an individual character of the String object can be converted to a `char`

Method	Description
`charAt(int index)`	Returns a `char` value that corresponds to the letter position index
`toCharArray()`	Returns the String object converted to a `char` array

- Recall that `char` variables are actually stored using the Unicode representation of the letter
- Letters A to Z have values 65 to 90
- Lowercase letters a to z have values 97 to 122
- `char` values can be compared with relational operators such as ==, < or >
  - `If (letter1 > letter2)...`

ICS4U Module 5: Note + Exercise 1b

Example Program: CountLetters

```
/*
 * CountLetters.java from Module 5
 * Count the occurences of letters in a string.
  */

 /**
  * The occurences of letters in a string are counted.
  */

 import java.util.Scanner;

 public class CountLetters {

      public static void main(String[] args) {
            final int LOW = 'A';           //smallest possible value
            final int HIGH = 'Z';          //highest possible value
            int[] letterCounts = new int[HIGH - LOW + 1];
            Scanner input = new Scanner(System.in);
            String word;
            char[] wordLetters;
            int offset;       //array index

            /* prompt user for a word */
            System.out.print("Enter a word: ");
            word = input.nextLine();

            /* convert word to char array and count letter occurrences */
            word = word.toUpperCase();
            wordLetters = word.toCharArray();
            for (int letter = 0; letter < wordLetters.length; letter++) {
                  offset = wordLetters[letter] - LOW;
                  letterCounts[offset] += 1;
            }

            /* show letter occurrences */
            for (int i = LOW; i <= HIGH; i++) {
                  System.out.println((char)i + ": " + letterCounts[i -
LOW]);
            }
      }
}
```

first the char array wordLetters is declared

now the array is initialized

i must be cast as a character here to produce labels for the contents of the array (turns it into Unicode)

Output:

```
Enter a word: algorithm
A: 1
B: 0
C: 0
D: 0
E: 0
F: 0
G: 1
H: 1
I: 1
K: 0
L: 1
M: 1
N: 0
O: 1
P: 0
Q: 0
R: 1
S: 0
T: 1
U: 0
V: 0
W: 0
X: 0
Y: 0
Z: 0
```

Programming Exercise:

a) The LetterCount application is limited to counting letters in a single word. Modify the LetterCount application to count the letters in an entire phrase, which contains spaces. Care must be taken to ignore the spaces and any other non-alphabetic character found in the phrase. Be sure to change comments and variable names appropriately so that the reader of the application code understands that the letters in a phrase are counted.

b) Create a BackwardsName application that prompts the user for his or her name and then displays the name backwards

Submit your source code for both exercises to the Google Doc "ICS4U – Activity Submission Form"