Name _____ Period _____

# Processing Text with Arrays

## Your Tasks (Mark these off as you go)

- ☐ Write code to populate an array from the command line
- ☐ Write code to count all the words in text passed to the command line
- ☐ Write code to calculate the average length of each word in text passed to the command line
- ☐ Write code to count the number of each letter in text passed to the command line
- ☐ Write code to count all the words of each length in text passed to the command line
- ☐ Receive credit for this lab guide

## ☐ Write code to populate an array from the command line

In our previous lesson we learned that we have been declaring arrays since we first started writing java programs. Every time you write a main method you declare the array, *String args[]*.

Write a driver class called ProcessText. Then write the main method. In the main method declaration, declare an array called text, instead of *args*

```
public class ProcessText{
    public static void main(String text[]){
    }
}
```

Write code that code be used to compile the class above. Then write code that could be used to both run and pass the following text to your program, "This is a test file for the second programming project in AP Computer Science A! This is a word in (parentheses) and here is another word with some w-i-e-r-d formatting and here is my lucky number, 777!"

```
javac ProcessText.java
java ProcessText "This is a test file for the second programming project in AP Computer
Science A! This is a word in (parentheses) and here is another word with some w-i-e-r-d formatting and here is my
lucky number, 777!"
```

Write code that could be used to print the text that was passed to your program.

```
System.out.println(text[0]);
```

## ☐ Write code to count all the words in your program

The text passed to the command line above has lots of weird formatting. Write code that could be used to count all the words in the program. For this task you can assume that spaces separate the words.

```
Scanner wordSC = new Scanner(text[0]);
int count = 0;
while(wordSC.hasNext()){
    count++;
}
```

## ☐ Write code to calculate the average length of each word in text passed to the command line

Now that you have determined how to count all the words, write code that could be used to determine the average word length of all the words in the text.

```
Scanner wordSC = new Scanner(text[0]);
int count = 0;
int totalLength = 0
while(wordSC.hasNext()){
    count++;
    String currentWord = wordSC.next();
    totalLength += currentWord.length();
}
double avgWordLength = totalLength/count;
```

How might you modify your loop above to determine the longest word in the text?

```
Scanner wordSC = new Scanner(text[0]);
int count = 0;
int totalLength = 0
String longest = wordSC.next();//let's assume the first word is the longest
while(wordSC.hasNext()){
    count++;
    String currentWord = wordSC.next();
    totalLength += currentWord.length();
    if(currentWord.length() > longest.length(){
        longest = currentWord;
    }
}
```

## ☐ Write code to count the number of each letter in text passed to the command line

In the space below, write code that could be used to count the number of occurrences of each letter, a, b, c, d, etc. The number of each letter must be stored in an array called int letterCount[]. In this array, index 0 should represent the "A's", index 1 should represent the "B's", etc. How might you do this without using if statements?

```
Scanner wordSC = new Scanner(text[0]);
int count = 0;
int letterCount[] = new int[26];
while(wordSC.hasNext()){
    String currentWord = wordSC.next().toUpperCase();
    for(int i = 0; i < currentWord.length();i++){
        if(currentWord.charAt(i) >= 65 || currentWord.char(i) <=90){
            //We know it's a letter... so now what
             int letter = (int)currentWord.charAt(i) - 65;
             letterCount[letter]++;
         }
     }
}
```

## ☐ Write code to count all the words of each length in text passed to the command line

In the space below, write code that could be used to count the frequency of each word length. The occurrences of each word length should be stored in an array called int wordLengthCount[]. In the wordLengthCount[] array, index 1 should represent the number of words with a length of 1, index 2 should represent the number of words with a length of 2, etc.

```
Scanner wordSC = new Scanner(text[0]);
int count = 0;
int letterCount[] = new int[26];
int wordLengthCount[] = new int[50];
while(wordSC.hasNext()){
    String currentWord = wordSC.next().toUpperCase();
    for(int i = 0; i < currentWord.length();i++){
        if(currentWord.charAt(i) >= 65 || currentWord.char(i) <=90){
            //We know it's a letter... so now what
             int letter = (int)currentWord.charAt(i) - 65;
             letterCount[letter]++;
        }//end if
     wordLengthCount[currentWord.length()]++;
    }
}
```

## ☐ Receive credit for this lab guide

Submit this portion of the lab to Pluska to receive credit for the lab guide. Once received, your completed code challenges will also be graded and will count towards your final lab grade.