

Programs for this lab should be stored in your Eclipse repository. Remember to use a location on your P: drive or a USB drive (not the C: drive). Create a new Java project in your workspace called *lab7*. When your programs are running correctly, **turn in a printout** of the Java code and **send an email** containing your .java files for each program in lab7 (as separate file attachments) to the instructor at stlauterburg@salisbury.edu with the subject line “COSC 117 Lab 7”.

Problem 1

Write a program called VowelCount that does the following:

1. Ask the user to type in a sentence. The Scanner method `nextLine()` will accept an entire line and return it as a String. This is different from `next()` which will accept only a single ‘token’.
2. Count the number of *lower-case* vowels in the sentence. For purposes of this exercise consider the characters 'a', 'e', 'i', 'o' and 'u' to be vowels.
3. Print out the count for each vowel.

Example:

```
Please enter a sentence:
Your eyes are blue.
```

```
Number of a's: 1
Number of e's: 4
Number of i's: 0
Number of o's: 1
Number of u's: 2
```

Problem 2

Write a program called Combinations that prompts the user for two integer numbers: the number of objects in a set (n) and the number of objects to be chosen from that set (k). The program should then call a function that determines the number of ways that k objects can be selected from n distinct objects.

The combinations function $C(n, k)$ can be calculated in terms of factorials (see <http://en.wikipedia.org/wiki/Combination>):

$$C(n, k) = \frac{n!}{k! * (n-k)!}$$

Your program should contain three methods:

- 1) a **main** method which does all input and output (in other words, you don't need a separate input method for *n* and *k*; just prompt for the values in the main method)
- 2) a method called **factorial** that computes the factorial of a number (you can use either a for loop or a while loop)
- 3) a method called **combinations** that computes the number of combinations. The combinations method should be relatively short (around 5 lines). It simply calls the factorial function with an appropriate argument when it needs a factorial computed (hint:

it will do this three times).

Example 1:

Enter the number of items to choose (k): **3**

Enter the number of items (n): **5**

There are 10 combinations.

Problem 3

In the game Scrabble, words are scored by summing the point value for each letter in the word. A letter's point value is inversely proportional to its frequency in English words. Letters are awarded points as follows:

1 point: **A, E, I, L, N, O, R, S, T, U**
2 points: **D, G**
3 points: **B, C, M, P**
4 points: **F, H, V, W, Y**
5 points: **K**
8 points: **J, X**
10 points: **Q, Z**

Write a program called ScrabbleScore that accepts a word from the user and uses an if-else **statement** to compute its score (not counting any of the bonuses that might normally occur in the game). Do *not* assume that the user enters words all in one case. *Hint:* think about converting the word to all lower- or upper-case before processing its score. Take a look at <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>. The program should display the word as originally entered along with the number of points it is worth.

Example 1:

Please enter a word: **FARM**
The word FARM is worth 9 points.

Example 2:

Please enter a word: **Farmer**
The word Farmer is worth 11 points.