C++ code kata: Week #1

Hello 👋

Ground statement

Consider you as a data engineer. As a data engineer, your first task is to make vowel recognition dataset. For the exercise, you have to write `vowels_counter()` that will take a string and returns the total number of vowels in it. e.g. 👇

```
auto nCount = vowels_counter("abcde")
printf("%d", n Count)
2
```

Surface #1

If you have completed **Ground Statement**, then for **surface # 1** exercise you have to return the sum of the total number of vowels till it is a single digit. e.g. 👇

```
auto nCSum = vowels_counter("A cup of tea and coffee with sugar cube.")
printf("%d", n Count)
5
```

Surface #2

To reach **surface # 2** you have to return the sum of the total number of vowels in each substring. "abba" will have substrings "a, ab, bb, ba, abb, bba, abba", and the number of vowels in each substring will be 1 1, 0, 1, 1, 1, 2 and the total number will be sum of all vowels presence e.g. 👇

```
auto nCSum = vowels_counter("abba")
printf("%d", n Count)
7
```

Tools

To write solutions you can download tools from:

Tool	Usage	Download Location
Visual Studio Code	Lightweight program editor.	Download
GCC (Linux)	C / C++ Compiler.	Download
MingGW (Windows)	C / C++ Compiler.	Download
XCode toolkit (MacOS)	IDE with multi language support.	Download
Catch2	C++ Test framework	Download

Test Script

Test Script for this week's problem can be downloaded from **here**.

Sensai Says

"What you learn is not what you read or listened to, but rather what you attempted at..."
Progressive learning

If you feel the exercise a little bit difficult to solve do not get disheartened. The whole idea behind these programming exercises is not to solve them but rather attempt them.

Try to attempt them in as many ways as possible, you will learn new techniques that will be very helpful to you in longer run especially in work field.

We will present you with **solution mail / document** also. The solution will show you various ways to solve a problem and why a technique is better than last one.

We encourage you to make notes from the solution provided and try to apply what you have learnt in future exercise.

What I will gain from these exercises ?

1. Better and faster way to solve an exercise.
2. Reusable components like containers, algorithms etc. that you can apply to problem at handrather than designign your own.
3. Confidence and attitude to solve a problem in new ways, instead of trying monotonus techniques.
4. Writing the robust and quality software usign **test driven development**.

All the best 👌

Still have some questions related to this exercise. Reach us at

1. 💬 programmingdays (Skype)