

# Kata-Log

## All Topics

### Choose Your Level

★ Starter

★ Experienced

### Choose a Topic

Agile

BDD

Golden Master

Mocks

Outside-In

Pair-Programming

Refactoring

SOLID Principles

Software-Design

TDD

### Choose a Constraint

Baby Steps

Ensemble Programming

Minimalist Coder

Mute Ping Pong

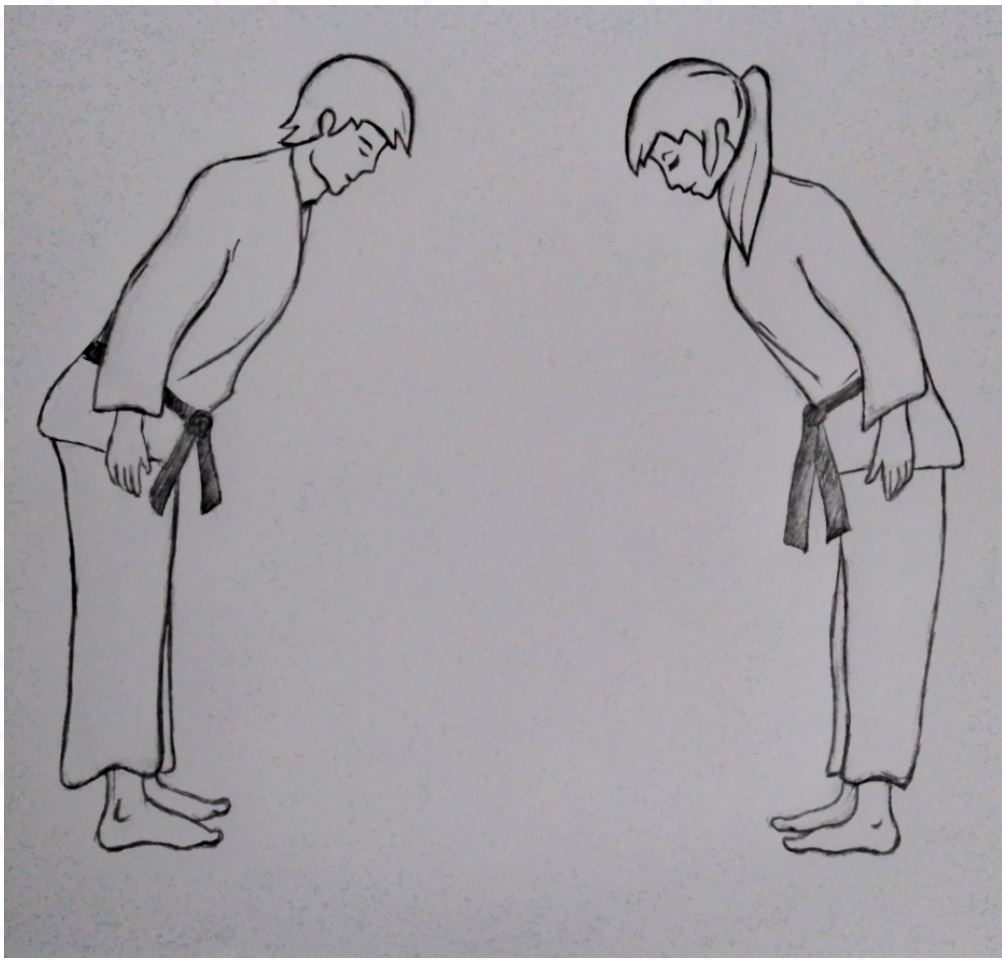
No Primitives

Simple Design

Tell! Don't ask!

The 70s Compiler

## String Calculator Kata



### Credits

Inspired by [Roy Oshero](#)

## Incremental Kata - no peeping ahead!

This is an incremental kata to simulate a real business situation: work your way through the steps in order, but do not read the next requirement before you have finished your current one.

### Step 1

Create a simple String calculator with a method signature:

```
int Add(string numbers)
```

The method can take up to two numbers, separated by commas, and will return their sum.

For example `""` or `"1"` or `"1,2"` as inputs.

For an empty string it will return 0.

### Step 2

Allow the Add method to handle an unknown amount of numbers.

### Step 3

Allow the Add method to handle new lines between numbers (instead of commas):

- The following input is ok: `"1\n2,3"` (will equal 6)
- The following input is NOT ok: `"1,\n"` (not need to prove it - just clarifying)

### Step 4

Support different delimiters:

- To change a delimiter, the beginning of the string will contain a separate line that looks like this: `"//[delimiter]\n[numbers...]"` for example `"//;\n1;2"` should return three where the default delimiter is `','`.
- The first line is optional. All existing scenarios should still be supported.

### Step 5

Calling Add with a negative number will throw an exception `"negatives not allowed"` - and the negative that was passed.

If there are multiple negatives, show all of them in the exception message.

### Step 6

Numbers bigger than 1000 should be ignored, so adding  $2 + 1001 = 2$

### Step 7

Delimiters can be of any length with the following format: `"//[delimiter]\n"` for example: `"//[***]\n1***2***3"` should return 6.

### Step 8

Allow multiple delimiters like this: `"//[delim1][delim2]\n"` for example `"//[*][%]\n1*2%3"` should return 6.

### Step 9

Make sure you can also handle multiple delimiters with length longer than one char.

### Starting Points

[C++](#), [C#](#), [Clojure](#), [D](#), [Elixir](#), [F#](#), [Go](#), [Haskell](#), [Java](#), [JavaScript](#), [Kotlin](#), [PHP](#), [Python](#), [ReScript](#), [Ruby](#), [Rust](#), [Scala](#), [TypeScript](#)

[Clojure](#), [CoffeeScript](#), [C++](#), [C#](#), [Erlang](#), [Groovy](#), [Intercal](#), [Java](#), [JavaScript](#), [Lisp](#), [PHP](#), [Ruby](#), [Scala](#)

### Image credits

Image by [Rachel M. Carmena](#). It represents a greeting of respect and gratitude made at the beginning and at the end of a kata.

Comprehensive **Kata-Log** made for facilitators and dojo lovers.  
All contents are [public domain](#) and can be used without any strings attached.

Questions? Additions? Improvements?  
Please [add a new kata](#) or [create an issue](#) .

Proudly hosted [on GitHub](#)