# Exercise: Regular Expressions

This document defines in-class exercise problems from the ["Advanced C#" Course @ Software University](http://softuni.bg/courses/advanced-csharp/). You are presented with some problems and certain steps you need to take in order to accomplish the tasks.

## Match Full Name

Write a regular expression to match a valid full name. A valid full name consists of **two words**, each word **starts** with a **capital letter** and contains **only lowercase letters afterwards**; each word should be **at least two letters long**; the two words should be **separated by a single space**.

To help you out, we've outlined several steps:

* Use an online regex tester like <https://regex101.com/>
* Check out how to use **character sets** (denoted with square brackets - "[]")
* Specify that you want two words with a space between them (the **space character ' '**, and not any whitespace symbol)
* For each word, specify that it should begin with an uppercase letter using a **character set**. The desired characters are in a range – **from 'A' to 'Z'**.
* For each word, specify that what follows the first letter are only **lowercase letters**, one or more – use another character set and the correct **quantifier**.
* To prevent capturing of letters across new lines, put "\b" at the beginning and at the end of your regex. This will ensure that what precedes and what follows the match is a word boundary (like a new line).

In order to check your regex, use these values for reference (paste all of them in the Test String field):

|  |  |
| --- | --- |
| **Match ALL of these** | **Match NONE of these** |
| Ivan Ivanov | ivan ivanov, Ivan ivanov, ivan Ivanov, IVan Ivanov, Ivan IvAnov, Ivan Ivanov |

## Match Phone Number

Write a regular expression to match a valid phone number from Sofia. A valid number will start with "+359" followed by the area code (2) and then the number itself, consisting of 7 digits (separated in two group of 3 and 4 digits respectively). The different parts of the number are separated by **either a space or a hyphen** ('-'). Refer to the examples to get the idea.

* Use quantifiers to match a specific number of digits
* Use a capturing group to make sure the delimiter is **only one of the allowed characters** **(space or hyphen)** and not a combination of both. Use the group number to achieve this
* Add a word boundary at the end of the match to avoid partial matches (the last example on the right-hand side)
* Ensure that before the '+' sign there is either a space or the beginning of the string

|  |  |
| --- | --- |
| **Match ALL of these** | **Match NONE of these** |
| +359 2 222 2222  +359-2-222-2222 | 359-2-222-2222, +359/2/222/2222, +359-2 222 2222  +359 2-222-2222, +359-2-222-222, +359-2-222-22222 |