	Unaware			**Beginner**			**Course graduate**			**Ideal (future vision)**
1. Computational thinking and algorithms	I don't know what's an algorithm or I don't know how to design them.			If I have enough time or help, I can design a simple algorithm. However, I often make a mistake or get stuck.			I can think computationally and express the steps of an algorithm in code.			I can devise and understand different approaches to solving a problem and compare their advantages, disadvantages, and effectiveness. CLRS is my favorite book.
2. Data structures	I don't know or can't use data structures.			I know basic data structures, such as a list, and I tried using them.			I know and can use a list, dictionary, and simple objects in Python.			I know in detail the implementation and internal representation of data structures. I understand the time and space complexity of the individual operations and consider it while programming.
3. Functions	I don't know how to divide code into functions.			I can write a simple function without a return value.			I can divide a problem into several independent functions, which employ their return values.			I suitably decompose the whole program into functions, my code is readable. My functions have a single responsibility and a clear contract.
4. Working with errors	I'm afraid of errors in code. I can't fix the code if it contains errors.			I can fix simple syntactic errors (such as a missing parenthesis), but I usually need help with other errors.			I understand error messages in Python and can individually fix common syntactic, semantic, and logical bugs.			I also understand someone else's code. Even if the source of the bug is complex, I can help my colleagues/classmates.
5. Documentation	I've never read an official documentation.			I've read an official Python documentation, but sometimes it's hard for me to navigate in it.			If I don't understand what a function does in Python, I find its official documentation, read it, and then I'm able to use the function correctly.			I know the criteria for a quality documentation and I wrote one myself.

Author: Valdemar Švábenský, version: 16.7.2018