By trying to understand the imperative programming paradigm which uses procedural abstraction rather than Data abstraction, I realized that there are more differences between the Imperative Programming and OOP other than the way to abstract data.

I found that there can be lexical, syntax, accessing data, etc. differences between languages.

**For example:** in C++ which is an Imperative programming language

A Boolean <-> bool, inpout <-> cin, output <-> cout

In addition, I found that using the imperative programming languages for a large program is very hard and challenging. Because of the low-level data abstraction with the Imperative programming languages a large program will need lot of procedures which can make the program difficult to read and of course slow.

* Which parts were easier:

In my opinion, the easiest part when learning a new language is that when one knows the concept of algorithmic (pseudocode/steps to solve a problem) it become only a matter of how to translate the ideas that one has on how to solve the problem in the language chosen.

* which parts were more challenging:

I will say that the difficult part that I noticed when learning a new language, is the fact that each paradigm and especially each language has it own concepts, key words and rules. That mean each time that we learn a new language we learn some different concepts from the languages we knew, which can become easily confusing at some point; especially at this era of revolution technologic (with lot of different languages).