- What is a Programming Language?

A language (like any other languages) containing vocabulary and grammatical rules to express a set of detailed instructions to computers.

Some are high level → closer to our understanding (C++, Java python and …) , easier to work with , slower (must be translated to low – level languages in order to be executed by computers)

Some are low level → mostly describes machine's functionality rather than our solutions or our abstract ideas (Assembly or any machine's code).

This course → only working with high level languages and algorithms (software engineering and computer science)

For more info about low level languages and computer's hardware → refer to computer hardware engineering and computer architecture courses

High level languages are divided to two groups:

One, Interpreted languages (Python)

Two, Compiled languages (C++ , Java)

> The term between them is somehow a bit vague, a language can be translated by both compilers and Interpreters, however they are mostly designed to be originally translated by one of the groups of compilers and interpreters.
>
> They both take human codes and translate them to machine's code. (high level to low level)
>
> But… Whats the difference?

/* this part has been copied directly from internet and requires some modifications to make it unique rather than a pitiful copy. :D

“In a compiled language, the target machine directly translates the program. In an interpreted language, the source code is not directly translated by the target machine. Instead, a *different* program, aka the interpreter, reads and executes the code.

Okay… but what does that *actually* mean?

Imagine you have a hummus recipe that you want to make, but it's written in ancient Greek. There are two ways you, a non-ancient-Greek speaker, could follow its directions.

The first is if someone had already translated it into English for you. You (and anyone else who can speak English) could read the English version of the recipe and make hummus. Think of this translated recipe as the *compiled* version.

The second way is if you have a friend who knows ancient Greek. When you're ready to make hummus, your friend sits next to you and translates the recipe into English as you go, line by line. In this case, your friend is the interpreter for the *interpreted* version of the recipe.”

*/

Pros and Cons:

Interpreted languages are slower, less efficient and give you less control over the memory and your processing units, however,

This could also mean they’re easier to use, because there is an interpreter helping you make your intention clearer to the computer.