simplified hierarchy

Certainly! Here's a simplified hierarchy of how source code is processed and executed in both programming and scripting languages:

Programming Language:
1. **Source Code**: You write the source code in a programming language (e.g., C++).
2. **Compilation**: The source code is processed by a compiler, which checks for errors, translates it into machine code, and produces an executable file.
3. **Executable**: The compiled code (executable) is generated, which can be run independently on the target platform.

Scripting Language:
1. **Source Code**: You write the source code in a scripting language (e.g., Python).
2. **Interpretation**: The source code is executed by an interpreter, which reads the code line by line, translates it to machine code on the fly, and performs the actions described in the code.
3. **Output**: The output or result of the script's execution is displayed or used as intended. Scripting languages often don't produce a standalone executable.

The key distinction here is that programming languages involve a separate compilation step, resulting in an independent executable file, while scripting languages are interpreted, and the code is executed directly without creating a standalone executable. This difference affects the way you run and distribute programs written in these languages.