*Low-Level vs. High-Level Languages**

- **Low-Level:** Closer to machine code, fast, efficient but complex and platform-dependent (e.g., Assembly, Machine Code).
- **High-Level:** Easier to use, portable, but slower (e.g., Python, Java, C++).

*Interpreted vs. Compiled Languages**

- **Interpreted:** Executes code line-by-line, more flexible but slower (e.g., Python, JavaScript).
- **Compiled:** Translates code before execution, faster but platform-dependent (e.g., C, C++).

*Programming vs. Scripted Languages**

- **Programming:** Used for full applications, statically typed, compiled (e.g., C, Java).
- **Scripted:** Used for automation, dynamically typed, interpreted (e.g., JavaScript, Python).

*Open Source vs. Proprietary Software**

- **Open Source:** Free access, community-driven, but may lack support (e.g., Linux, Python).
- **Proprietary:** Restricted access, secure but costly (e.g., Windows, macOS).

***OOP Support vs. Non-OOP**

- **OOP Support:** Modular, reusable, but complex (e.g., Java, C++).