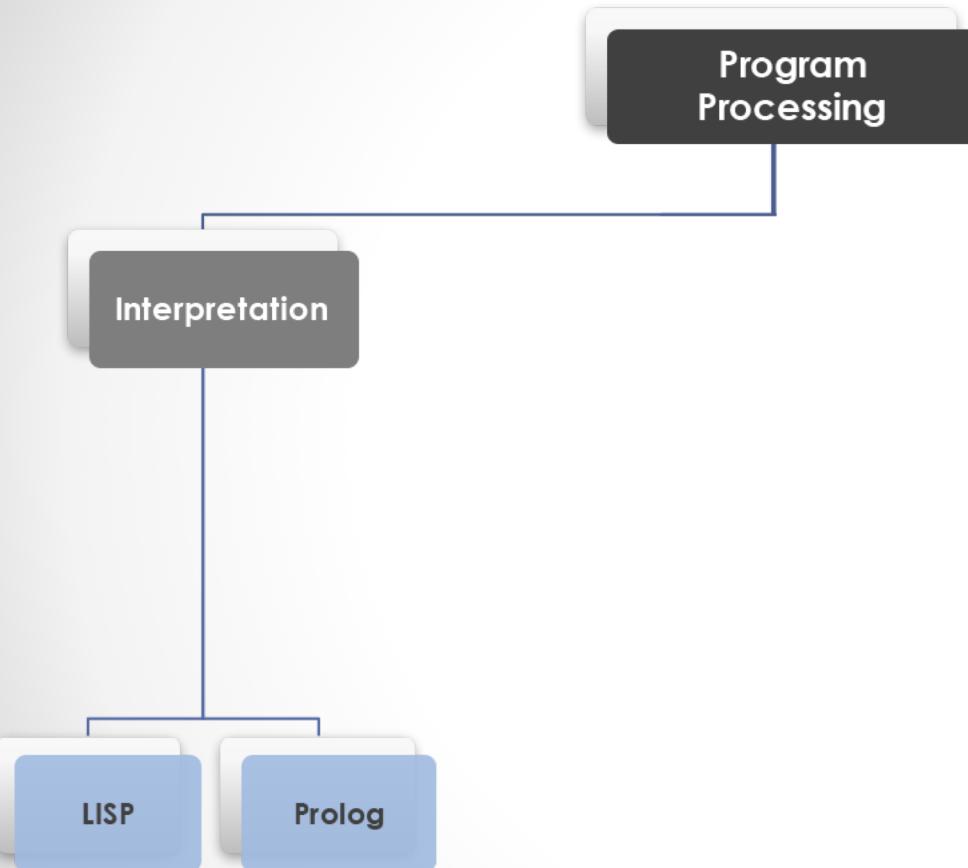


CSE240 – Introduction to Programming Languages (online)

Lecture 04:
Program Processing

Javier Gonzalez-Sanchez

javiergs@asu.edu
javiergs.engineering.asu.edu
Office Hours: By appointment



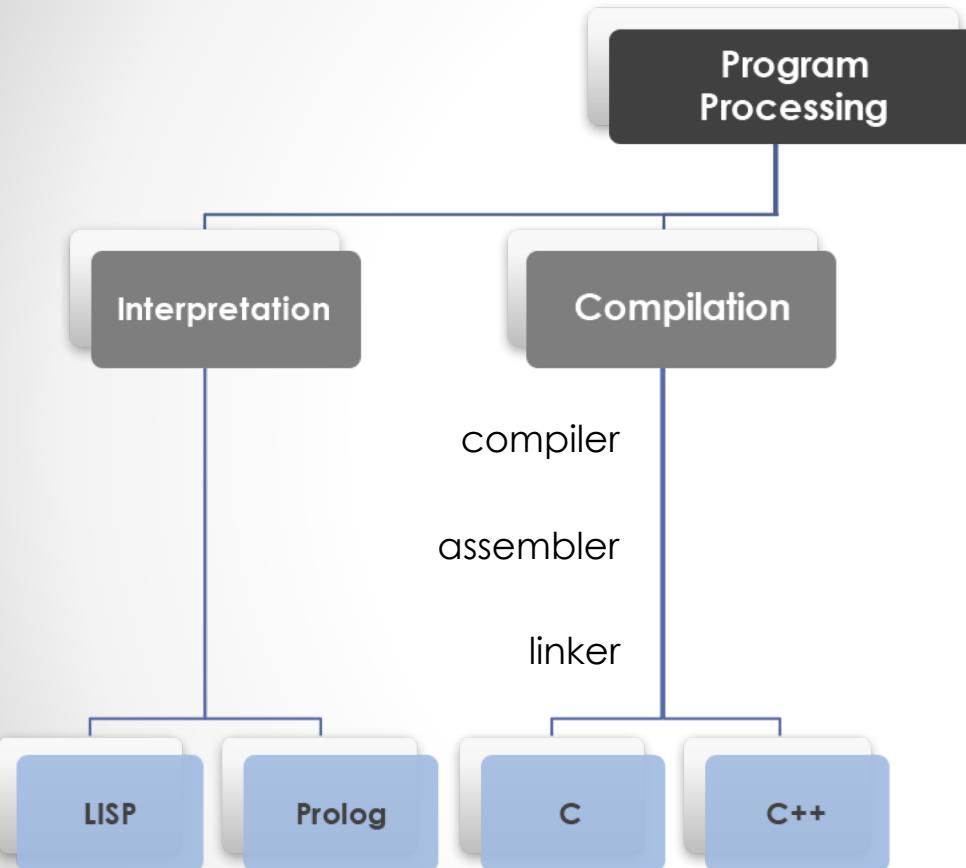
Interpreter is a program that translates in runtime and **executes** each statement in the **high level-language**.

Advantages:

- No separate compilation phase (quicker program development);
- Good debugging information since the source code is being executed.

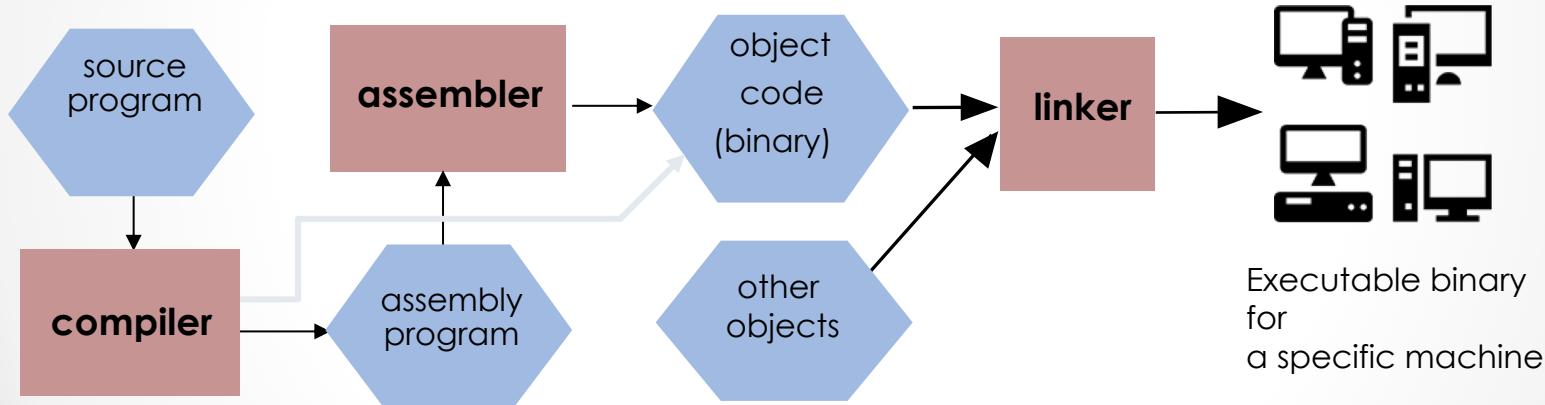
Disadvantages:

- Slow execution
- Can use more memory space



Program Processing: Compilation

Compiler is a program that translates the entire code from source **to** assembly code/machine code. **Linker** resolves external references (e.g., bring in code from libraries)



Executable binary
for
a specific machine

Program Processing: Compilation

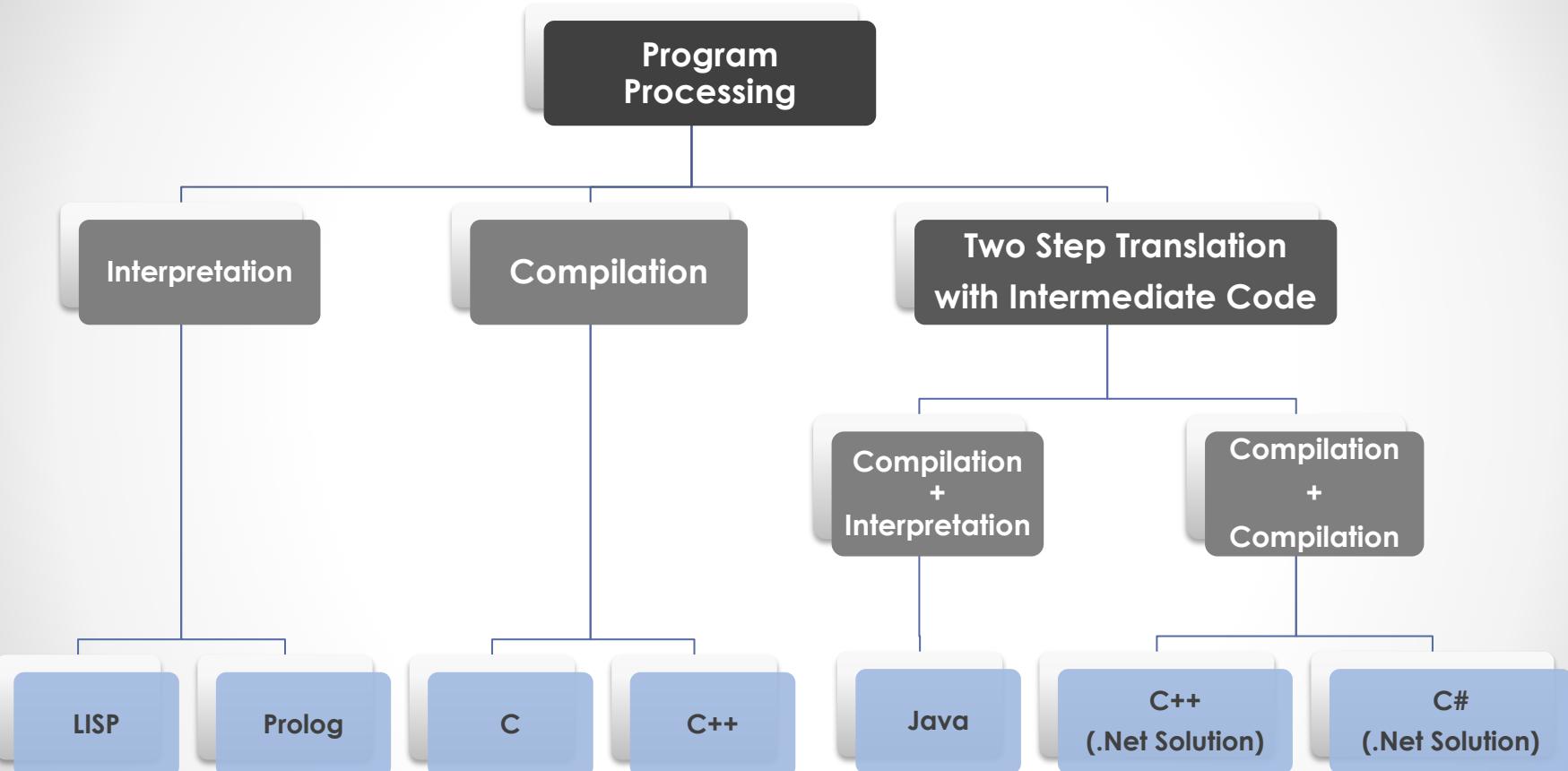
Compiler is a program that translates the entire code from source **to** assembly code/machine code. **Linker** resolves external references (e.g., bring in code from libraries)

Advantages:

- Faster than interpretation
- Good for multi-module programs

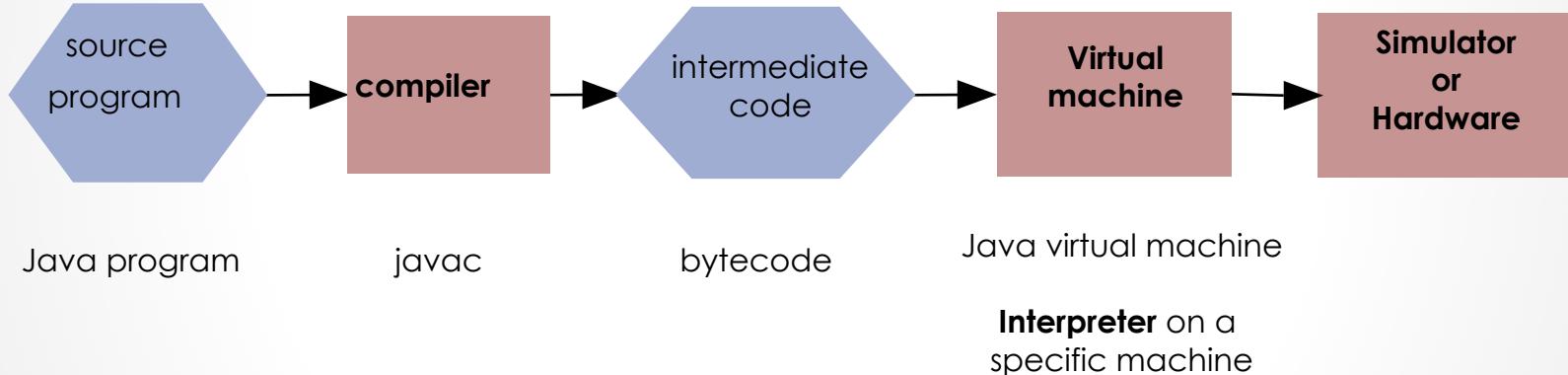
Disadvantages:

- Separate compilation phase
- May lose debugging info



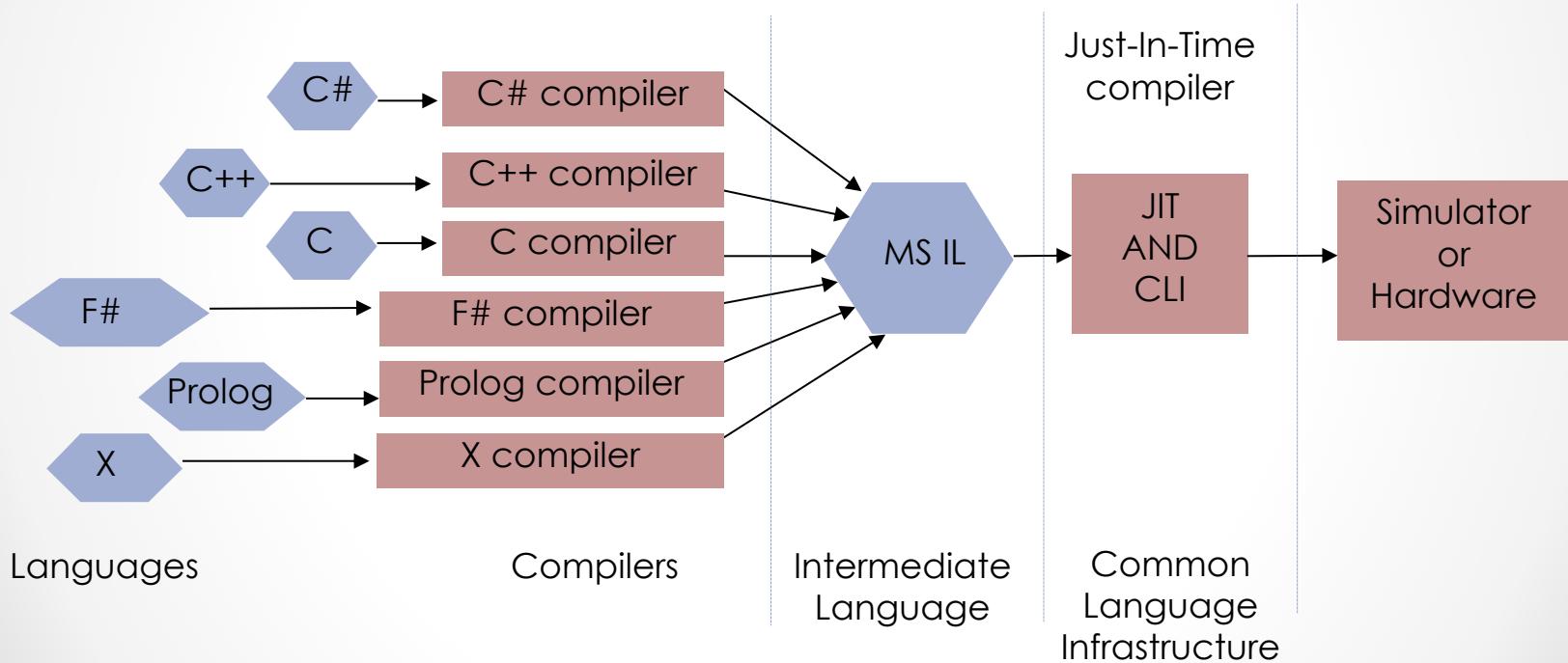
Program Processing: Intermediate Code (Java)

Intermediate code make the language and compiler machine-independent. This concept was implemented in 1970's to simplify the compiler design. It is used in Java.



Program Processing: Intermediate Code (Microsoft .Net)

Microsoft uses an **intermediate code** for its Common Language Runtime (CLR) environment (.Net Runtime Environment), which is independent of languages



Program Processing: Intermediate Code

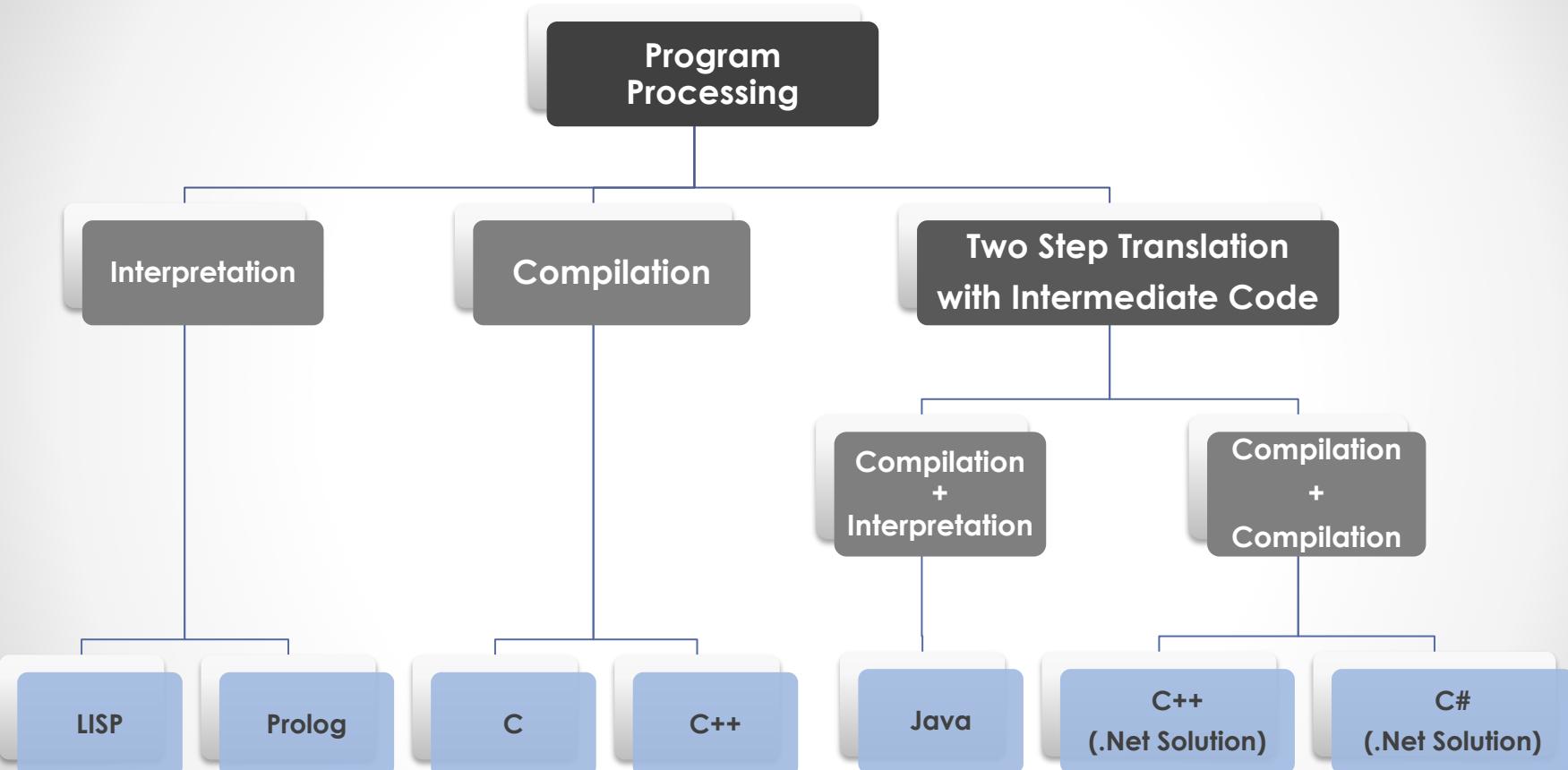
Intermediate code make the language & compiler machine-independent. This concept was implemented in 1970's to simplify the compiler design.

Advantages:

- A **single compiler for all machines**
- A small **virtual machine** (interpreter) that even fits in a web browser

Disadvantages:

- Slow execution and memory space
- Separate compilation phase and may lose debugging info





CSE240 – Introduction to Programming Languages (online)

Javier Gonzalez-Sanchez

javiergs@asu.edu

Fall 2017

Disclaimer. These slides can only be used as study material for the class CSE240 at ASU. They cannot be distributed or used for another purpose.