

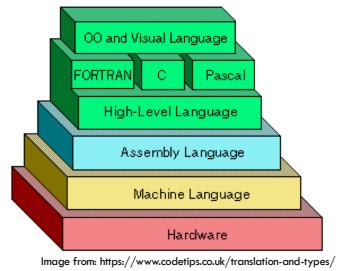
11 – LANGUAGE ABSTRACTIONS

COMP256 – COMPUTING ABSTRACTIONS

DICKINSON COLLEGE

LANGUAGE LEVELS AS ABSTRACTIONS

- Computers can only execute machine language programs.
 - All programs in all languages must ultimately reduce to a machine language program in order to be run.
 - for/if/while etc in a HLL program are abstractions for the equivalent machine language instructions that are actually executed.



HIGH/LOW LEVEL LANGUAGES

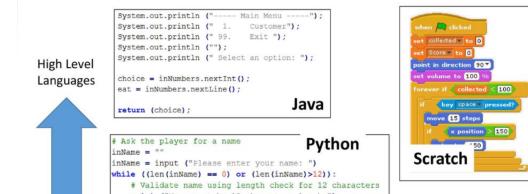
• High Level Languages (HLL)

- More human readable
- Abstract
 - Instructions focus on the task to be completed.
- Platform independent

• Low Level Languages (LLL)

- Machine readable (or close)
- Concrete
 - Instructions focus on machine capabilities.
- Platform dependent

C/C++, JavaScript, R, Perl, Go, Swift
Kotlin, PHP, SQL, C#, Matlab, Scala,
Fortran, Cobol, Ruby, Rust, Objective-C
Haskell, Scheme, Lisp, ...



Intel x86, IA-32, x86-64
ARM, MIPS, SPARC, DEC, IBM360,
Motorola 6800, ...
Assembly
Machine
00100101000111
001001000111

TRANSLATED VS INTERPRETED LANGUAGES

- Non-machine language programs are executed by being *translated* or *interpreted* *:

- **Translated Languages:** The entire source code (e.g. C/C++) for the program is converted to an equivalent machine language program that runs directly on the machine hardware.
- **Interpreted Languages:** The source code (e.g. JavaScript) for the program is read one instruction at a time and converted to an equivalent set of machine language instructions that are executed by the machine hardware.

* or both (more later)

TRANSLATED (COMPILED) HIGH LEVEL LANGUAGES

- In a *compiled high level language* another program, called a *compiler*, translates the HLL source code program into an equivalent machine language program for the target machine and operating system (OS).
 - The resulting machine language program is usually called the *executable program*.
- The *executable program* is run (*executed*) directly on the target machine hardware and OS.
 - Neither the source code nor the compiler are needed to distribute or run the program.
 - The executable program is specific to the target machine and OS.

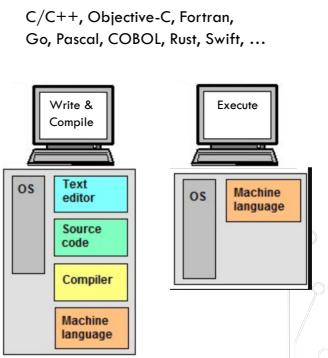


Image adapted from: <https://encyclopedia2.thefreedictionary.com/interpreter>

INTERPRETED HIGH LEVEL LANGUAGES

- In an *interpreted HLL language* an executable program, called the *interpreter*, reads the source code one instruction at a time and executes an equivalent set of machine language instructions.
 - The interpreter is a compiled machine language program.
- The source code is run by the interpreter.
 - The source code and an interpreter are required to run the program.
 - The program will run on any computer with an interpreter.

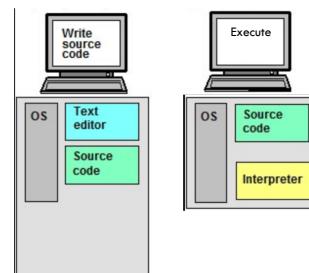


Image adapted from: <https://encyclopedia2.thefreedictionary.com/interpreter>

COMPILED VS INTERPRETED LANGUAGES

- Common Uses:**
 - Compiled:**
 - System programs (OS)
 - Large applications
 - Commercial application programs
 - Interpreted:**
 - Smaller programs
 - Task automation scripts
 - Web applications (e.g. JavaScript)

Compiled		Interpreted	
PROS	CONS	PROS	CONS
ready to run	not cross platform	cross-platform	interpreter required
often faster	inflexible	simpler to test	often slower
Source can be private	extra step	easier to debug	source code is public

Image adapted from: <https://learntocodewith.me/programming/source-code/>

VIRTUAL MACHINES (OR LET'S DO BOTH)

- Some languages are both translated and interpreted.
 - The source code is compiled to a *pseudo-machine language* called *byte-code*.
 - An executable program called a *virtual machine* (VM) interprets the byte-code.
- The byte-code is run by the virtual machine.
 - The byte-code and a virtual machine are needed to run the program, but the source code is not.
 - The byte-code program will run on any machine with a virtual machine program.

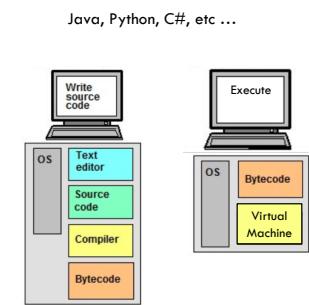
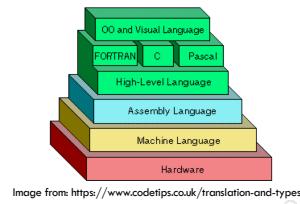


Image adapted from: <https://encyclopedia2.thefreedictionary.com/interpreter>

ASSEMBLY LANGUAGE PROGRAMMING

- An assembly language is an abstraction for a machine language.
 - Assembly language instructions:
 - Are mnemonic (human readable, easier to remember text) version of machine language instructions.
 - Usually have a 1-to-1 correspondence to machine language instructions.

Machine Language	Meaning	Assembly Language
1000 0001 0 00 00110	R0 = MM[5]	LOAD R2 5
1010 0001 00 11 00 01	R3 = R0 + R1	ADD R3 R0 R1



ASSEMBLY LANGUAGES

- Assembly Languages are low level translated languages.
 - A program called an assembler translates the assembly source code program into an equivalent executable machine language program for the target machine.
 - Assembly language source code is highly specific to the target machine.
- Used in applications that require:
 - Precise control of the system
 - Highly optimized performance
 - E.g. Operating systems, device drivers, real-time control systems, game engines

Intel x86, IA-32, x86-64
ARM, MIPS, SPARC, DEC, IBM360,
Motorola 6800, etc ...

