An assembly language is a low-level [programming language](https://techterms.com/definition/programming_language) designed for a specific type of [processor](https://techterms.com/definition/processor). It may be produced by [compiling](https://techterms.com/definition/compile) source code from a high-level programming language (such as [C/C++](https://techterms.com/definition/cplusplus)) but can also be written from scratch. Assembly code can be converted to machine code using an [assembler](https://techterms.com/definition/assembler).  Assembly language (or assembler language),[[1]](https://en.wikipedia.org/wiki/Assembly_language#cite_note-IBM_2014_ASM-1) often abbreviated asm, is also referred to as any [low-level programming language](https://en.wikipedia.org/wiki/Low-level_programming_language) in which there is a very strong correspondence between the instructions in the language and the [architecture's](https://en.wikipedia.org/wiki/Computer_architecture) [machine code](https://en.wikipedia.org/wiki/Machine_code) [instructions](https://en.wikipedia.org/wiki/Instruction_set_architecture). Because assembly depends on the machine code instructions, every assembler has its own assembly language which is designed for exactly one specific computer architecture. Assembly language may also be called symbolic machine code.

while

Machine language sometimes referred to as machine code or object code, machine language is a collection of [binary](https://www.computerhope.com/jargon/b/binary.htm) digits or bits that the computer reads and interprets. Machine language is the only language a computer is capable of understanding. It is also referred to as a [computer program](https://en.wikipedia.org/wiki/Computer_program) written in machine language [instructions](https://en.wikipedia.org/wiki/Instruction_set) that can be executed directly by a computer's [central processing unit](https://en.wikipedia.org/wiki/Central_processing_unit) (CPU). Each instruction causes the CPU to perform a very specific task, such as a load, a store, a [jump](https://en.wikipedia.org/wiki/Jump_instruction), or an [arithmetic logic unit](https://en.wikipedia.org/wiki/Arithmetic_logic_unit) (ALU) operation on one or more units of data in the CPU's [registers](https://en.wikipedia.org/wiki/Processor_register) or [memory](https://en.wikipedia.org/wiki/CPU_cache).

While

high-level programming language is a [programming language](https://en.wikipedia.org/wiki/Programming_language) with strong [abstraction](https://en.wikipedia.org/wiki/Abstraction_%28computer_science%29) from the details of the [computer](https://en.wikipedia.org/wiki/Computer). In contrast to [low-level programming languages](https://en.wikipedia.org/wiki/Low-level_programming_language), it may use [natural language](https://en.wikipedia.org/wiki/Natural_language) elements, be easier to use, or may automate (or even hide entirely) significant areas of computing systems (e.g. [memory management](https://en.wikipedia.org/wiki/Memory_management)), making the process of developing a program simpler and more understandable than when using a lower-level language. The amount of abstraction provided defines how "high-level" a programming language is. It is also a [programming language](https://www.webopedia.com/TERM/P/programming_language.html) such as [C](https://www.webopedia.com/TERM/C/C.html), [FORTRAN](https://www.webopedia.com/TERM/F/FORTRAN.html), or [Pascal](https://www.webopedia.com/TERM/P/Pascal.html) that enables a [programmer](https://www.webopedia.com/TERM/P/programmer.html) to write [programs](https://www.webopedia.com/TERM/P/program.html) that are more or less independent of a particular type of [computer](https://www.webopedia.com/TERM/C/computer.html). Such [languages](https://www.webopedia.com/TERM/L/language.html) are considered high-level because they are closer to human languages and further from [machine languages](https://www.webopedia.com/TERM/M/machine_language.html).