

Programming Languages

A programming language is an interface between the programmer and the computer. The interface is defined by the grammar that describes the syntax of how keywords, names and symbols are used to build constructions such as expressions and statements, and the semantics that describe what the computer should do for each construct.

Programs are processed by compilers and interpreters that translate the program into a form that can be executed by another program, or execute it immediately.

- **Syntax:** the form or structure of the expressions, statements, and program units

Syntax Example: simple C if statement

if (<expr>)

<true-statement>

else

<false-statement>

- **Semantics:** the meaning of the expressions, statements, and program units

Semantics Example: if the expression evaluated to true (non-zero) execute the true statement otherwise execute the false statement.

Semantics should follow from syntax, the form of statements should be clear and imply what the statements do or how they should be used.

- Syntax and semantics provide a language's definition

Describing Syntax

- A *sentence* is a string of characters over some alphabet
- A *language* is a set of sentences
- A *lexeme* is the lowest level syntactic unit of a language (e.g., *,+,=, sum, begin)
- A *token* is a category of lexemes (e.g., identifier)

Algorithm

An algorithm is a sequence of well-defined instructions for completing a task or solving a problem. It can be described in a natural language, pseudocode, a flowchart, or even a programming language.

Properties of an Algorithm

An algorithm must possess the following properties:

finiteness: The algorithm must always terminate after a finite number of steps.

definiteness: Each step must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case.

input: An algorithm has zero or more inputs, taken from a specified set of objects.

output: An algorithm has one or more outputs, which have a specified relation to the inputs.

effectiveness: All operations to be performed must be sufficiently basic that they can be done exactly and in finite length.

Problems vs Algorithms vs Programs

- _ For each problem or class of problems, there may be many different algorithms.
- _ For each algorithm, there may be many different implementations (programs).

