
Declaration Order

Can an expression E refer to any name declared in the current scope, or only to names that are declared *before* E in the scope.

forward reference: an attempt to use a name before its declaration. A forward reference is a static semantic error.

C++ and Java relax the rules by dispensing with the define-before-use requirement in many cases. In both languages, members of a class (including those that are not defined until later in the program text) are visible inside all of the class's methods. In Java, classes themselves can be declared in any order.

Python takes the "whole block" scope rule one step further by dispensing with variable declarations altogether. In their place it adopts the convention that the local variables of subroutine S are precisely those variables that are written by some statement in the (static) body of S . If S is nested inside of T , and the name x appears on the left-hand side of the assignment statements in both S and T , then the x 's are distinct: there is one in S and one in T . Non-local variables are read-only unless explicitly imported (using Python's `global` statement).

Declarations and Definitions

Recursive types and subroutines introduce a problem for languages that require names to be declared before they can be used: how can two declarations each appear before the other? C and C++ handle this problem by distinguishing between the *declaration* of an object and its definition.

A **declaration** introduces a name and indicates its scope, but may omit certain implementation details.

A **definition** describes the object in sufficient detail for the compiler to determine its implementation.

If a declaration is not complete enough to be a definition, then a separate definition must appear somewhere else in the scope.

Nested Blocks

In many languages, local variables can be declared not only at the beginning of any subroutine, but also at the top of any `begin ... end({...})` block. Other languages, including Algol 68 and C, and all of C's descendants, are even more flexible, allowing declarations wherever a statement may appear. In most languages a nested declaration hides any outer declaration with the same name.