

## Representing Scope Information :

(9)

Every name possesses a region of validity within the source program called as scope of that name. Most languages have facilities for defining names with limited scopes. Two canonical examples are FORTRAN, where the scope of a name is a single subroutine and ALGOL, where the scope of a name is the block or procedure in which it is declared.

The rules governing the scope of names in block structured language are :

- a) A name declared within block B is valid only within B.
- b) If block  $B_1$  is nested within  $B_2$  then any name valid for  $B_2$  is also valid for  $B_1$  unless the identifier for that name is re-declared in  $B_1$ .

This situation allows the possibility that in the same program the same identifier may be declared several times as distinct names, possibly with different attributes and usually with different intended storage allocations. It is thus the responsibility of the symbol table to keep different declarations of the same identifier distinct. The usual method of making the distinction is to give a unique number to each program element that may have its own local data. The number of the currently active subprogram is computed by semantic rule associated with productions that recognize the beginning and end of a subprogram. The representation of the name inside the symbol table is a pair consisting of the corresponding identifier and