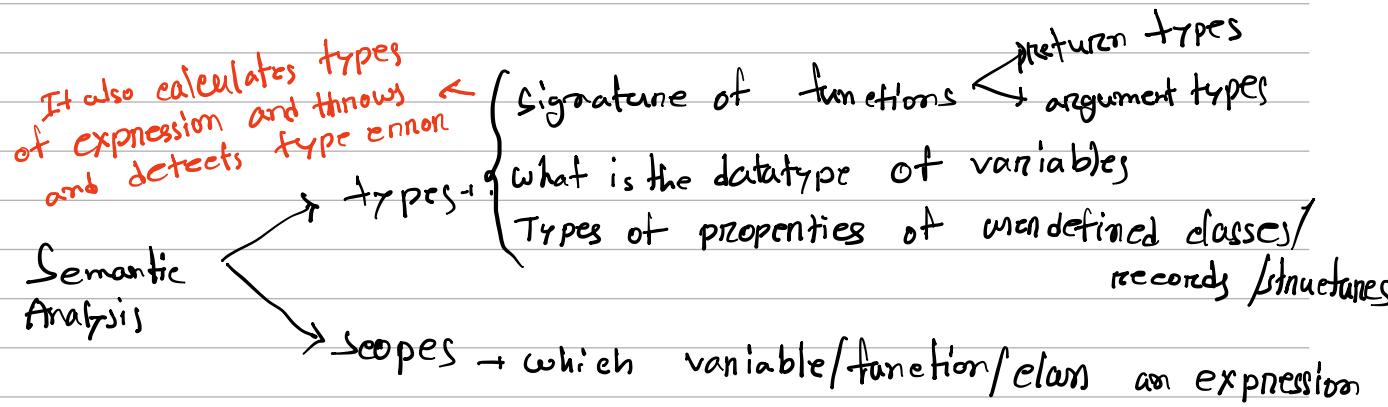


Semantic analysis

focus on meaning

Till now there is no spelling & grammatically mistake.

But still the meaning can be non-cohesive.



for example
 addition of int &
 float will be float,
 float cannot have
 modulus operation etc.

* Semantic analysis does work statement by statement.

If the whole program is grammatically correct
 then only syntax analyzer declares it correct.

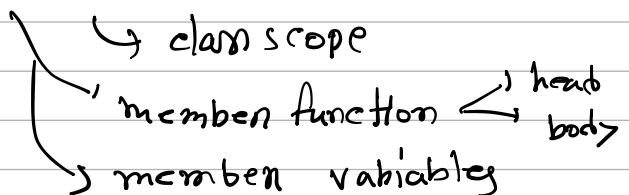
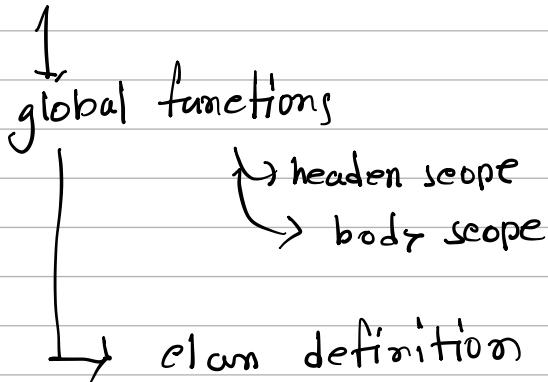
But semantic analyzer does not work like that
 since one expression's value can depend on
 another.

* Types & Scope related informations are stored in the symbol table.

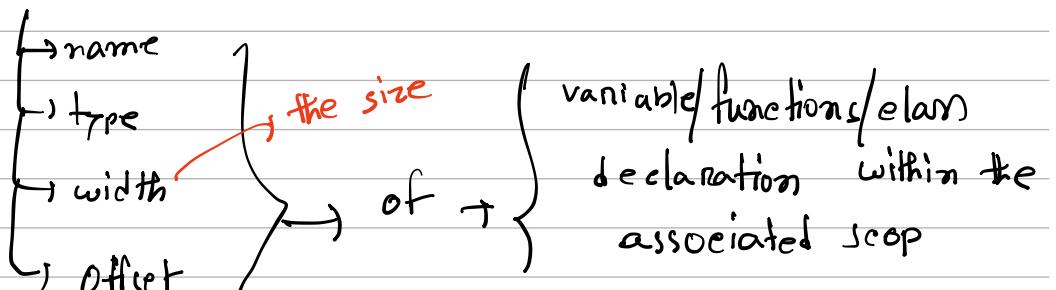
* For each scope there is a different symbol table.

function header scope → defined in function parameters
" body scope " " " " body

global scope



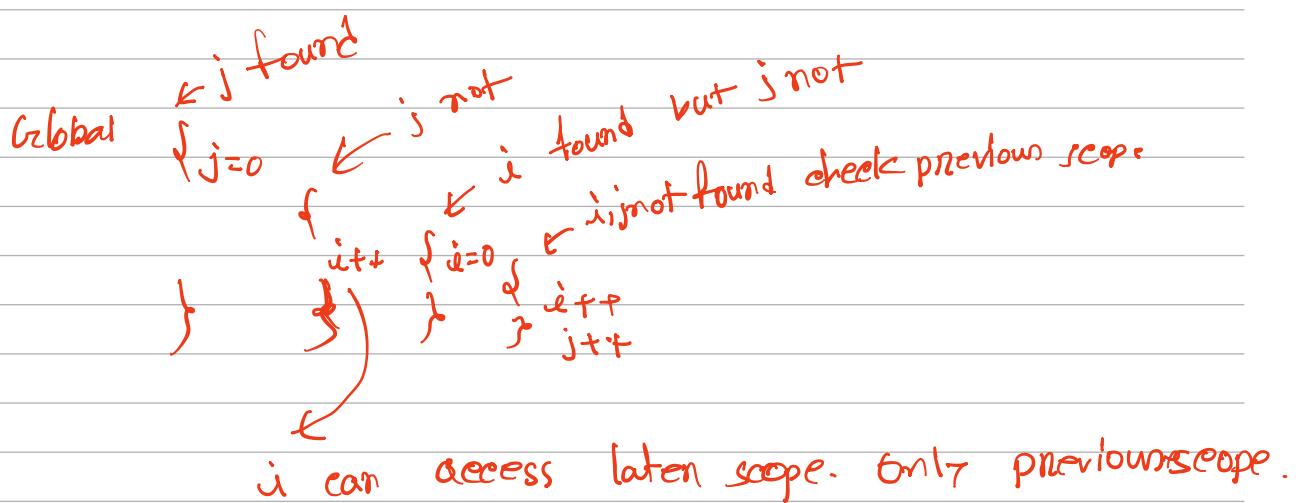
Symbol Table



assuming the logical address

It must occupy some memory

↳ program always starts from 0th and think whole RAM is empty. But the offset fixes the physical address for that program.

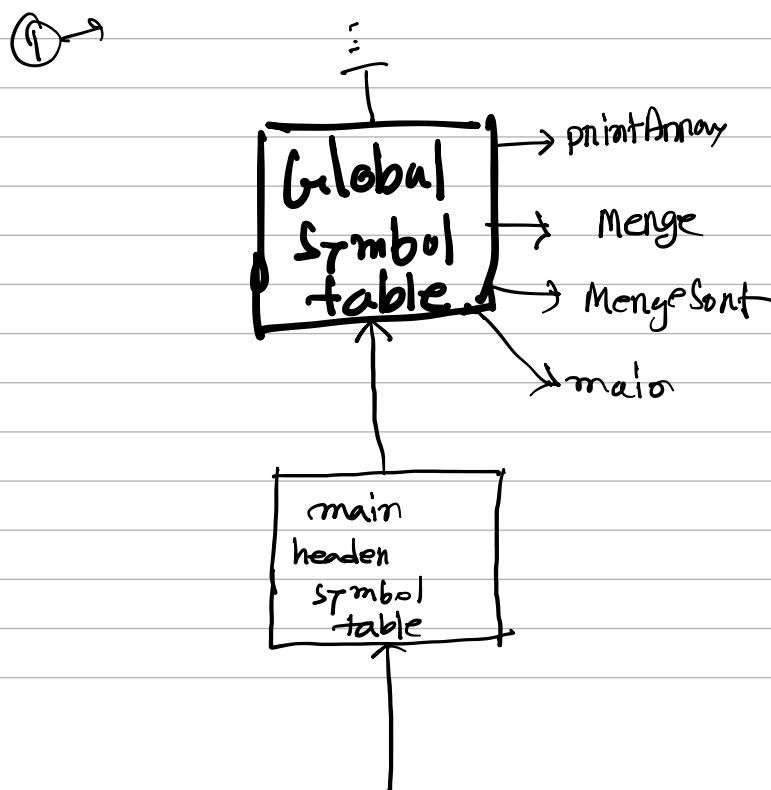


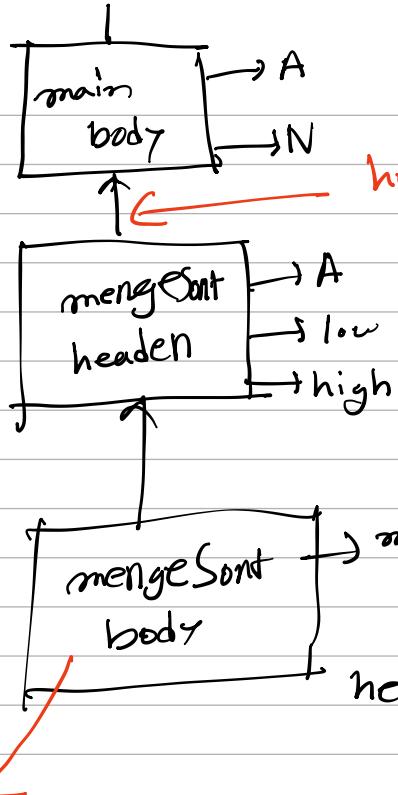
For all the scopes there will be different symbol table.

* Two ways to store the symbol table.

① Linked list of hash tables (Common)

② Hash table of Linked lists



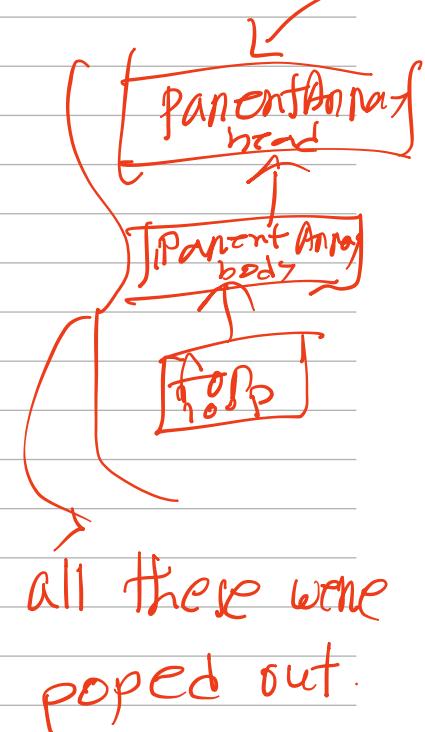


here in main body
printANarray was
here but when
we ended that
function the states
are dropped from
the stack

mid → for the line
if (low < high) {

from here
if we want
A we will
search here

then the previous
then till global.



all these were
popped out.