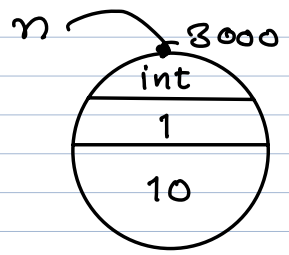
Case-III : REASSIGNMENT:

n = 10 ——— stmt-1

n = 15 ——— stmt-2.



Name	Addr
n	3000

└——— AFTER STATEMENT - (1).

---

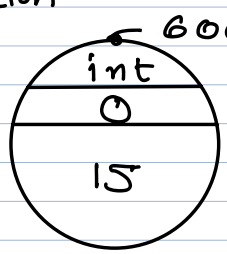
n = 15

Execute RHS.

Evaluate RHS = 15

Type Inference = class int

Memory Allocation



After RHS

Before LHS.

How to assign to LHS.

Step-I : Search LHS variable name in symbol table.

If the search fails then Goto Step-II [Definition Case]

If the search succeeds then Go to Step-III [Reassignment case]

Step - II : [Definition Case]

- Create a new symbol table entry.
- Keep variable name on LHS in the 'Name' column of the entry.
- Keep address of object obtained from evaluation of RHS in 'Address' column.
- Increase Ref. count of that object by - 1.

END

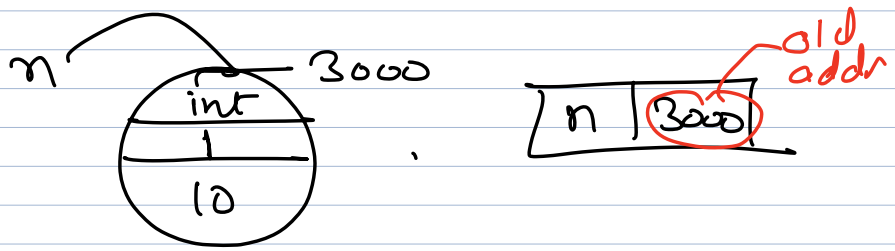
Step - III : [Reassignment Case]

Symbol table entry for LHS variable name is found.

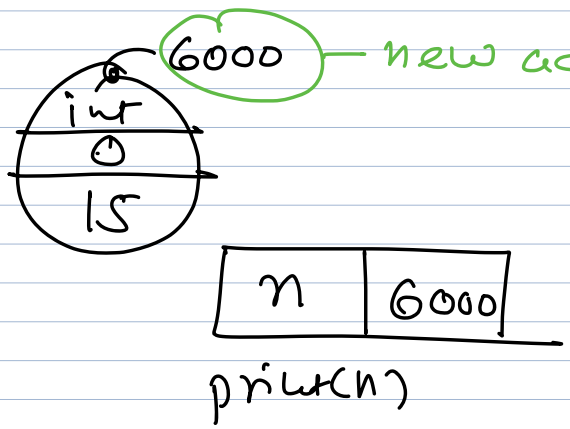
- Mark 'existing address' in the 'Address' column of entry as 'old address'
- Mark address of object obtained from the execution of RHS of the current stmt. 'new address.'
- Decrement the ref count of object located at old addr. by 1. If it falls down to zero after decrementing then invoke the garbage collector to free the memory given the object.
- Increment the ref. count of object located at 'new addr.' END.

Concrete case.

$n = 10$

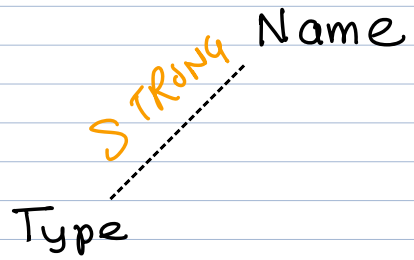


$n = 15$

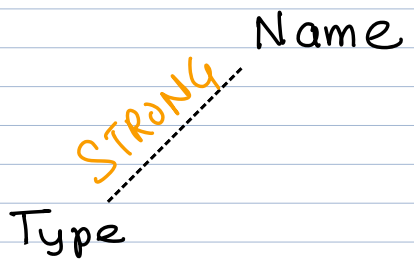


del v

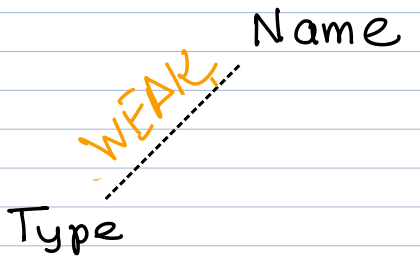
1. Search symbol table for v
2. If not found raise an exception
3. If found, decrement the ref. count of the associated object by 1.
   If it becomes 0 after decrement then release memory given to an object.
4. Remove the entire symbol table entry.



Instance variable of C, C++, Java, C#



Pointer in C/C++
Ref. C++/Java/C#



Ref. Var in Python

SUBJECTIVE