

Declarations in C

declaration - specifiers declarators ;

comma separated

list of declarators

(variables and/or
functions)

Declarations in C

declaration-specifiers declarators ;

int x ;

char *y , z ;

const unsigned int w ;

}
examples

Declarations in C

declaration-specifiers declarators;

3 categories of them

- storage class
- type qualifier
- type specifier

Storage classes

auto, static, extern
and register

- only one storage class can be specified
- comes first in declaration

Storage classes

properties of variables

- Storage duration
- scope
- linkage

Storage classes

properties of variables

- Storage duration
 - ↳ automatic or static
- scope
- linkage

Storage classes

properties of variables

- Storage duration
 - ↳ automatic or static
- scope
 - ↳ block or file
- linkage

Storage classes

properties of variables

- Storage duration
 - ↳ automatic or static
- scope
 - ↳ block or file
- linkage
 - ↳ external, internal, no linkage

- auto (by default variables are auto)
- static (we have discussed this earlier in the term)
- extern (says that this variable will be defined somewhere else; likely in another file)

- Register Storage class

register int x j

This one is a bit different oo

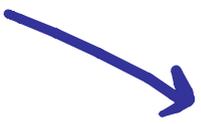
Storage Class for Functions

extern & Static {only two}

specifies linkage of the function (external or internal)

Type Qualifiers

const and volatile



value cannot
change.
It is constant.



value may
change (outside
of our control)

Note : const can appear
in function arguments

void foo (const char * word) ;

const char * word
we are not allowed to
modify the memory that
word points to

Type Specifiers

void , char , int ,
short , long , float ,
double , signed , unsigned

(struct , enum , union , typedef)

Note : storage class must
come first in
declaration

for type qualifier and
type specifier it doesn't
matter which order
they appear

Declarators

each consist of

- identifier

- possibly preceded by *

↗ pointer

- possibly followed by [] or ()

↗ array

↗ function