

Storage Classes

auto
✓

extern
✓

Static

register

3) Static

(global / fun)

local

File 2.c

extern int X;

File 1.c

static int X;

(Linker error)

Static + func \implies Private

Static + local

\simeq Global

Mem Seg

(data & bss)

Init

Zero

Life time

Program

Scope

{ }

hint

```
func() {
```

```
    static int X = 5;
```

```
}
```

```
main() {
```

```
    func();
```

Static

local

\Rightarrow

"lifetime"

Global

\Rightarrow

"SCOPE"

hint

Global & Static $\begin{cases} \text{local} \\ \text{Global} \end{cases}$

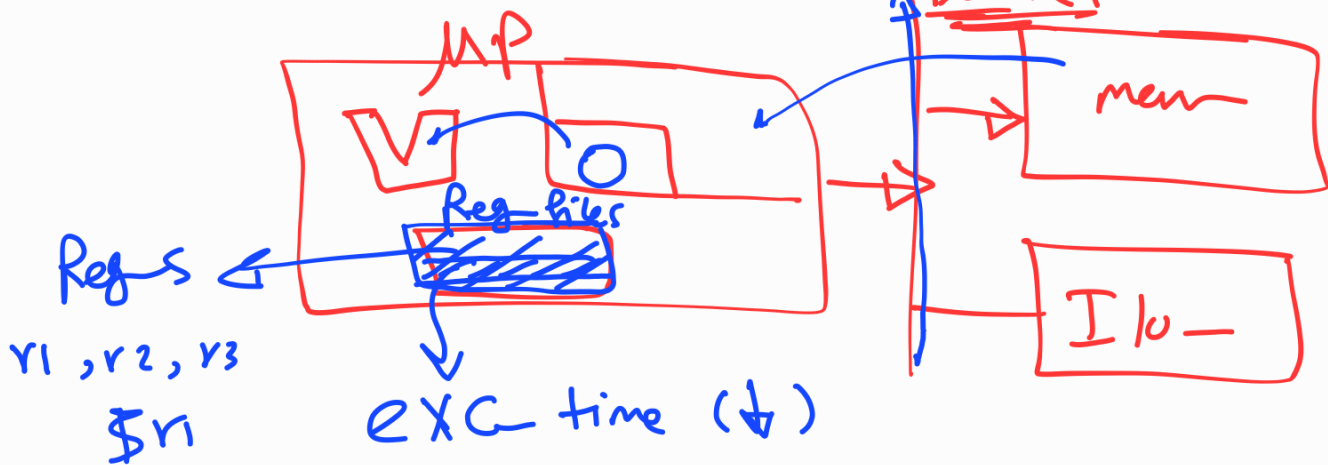
Init -

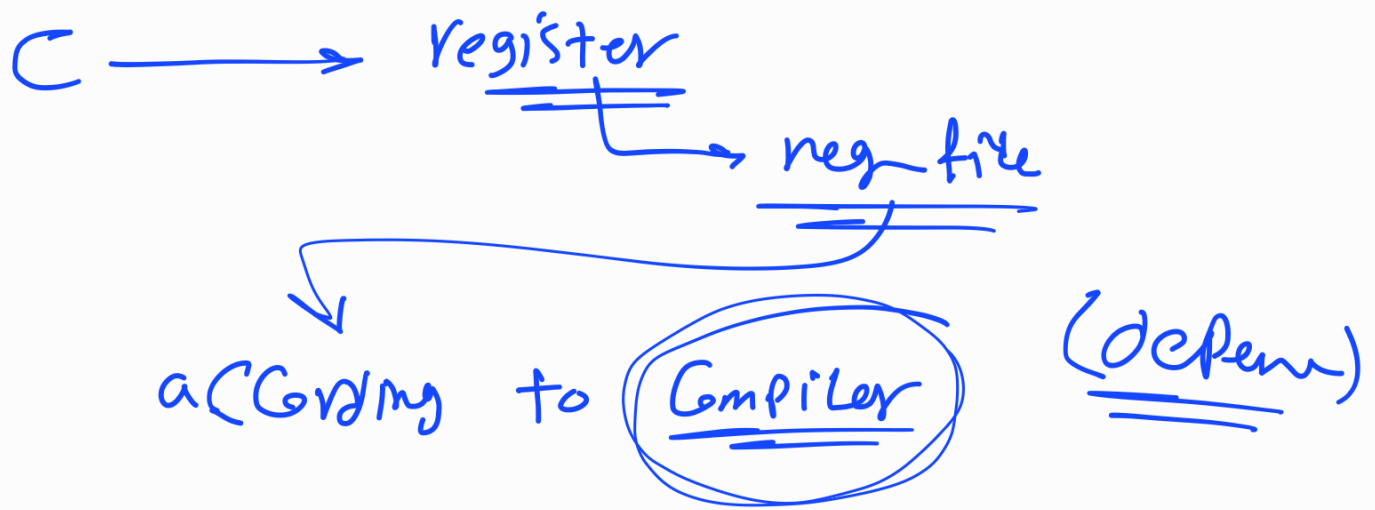
\Rightarrow

Const - value

Static int X = y; (X)

[4] register \Rightarrow (MP)



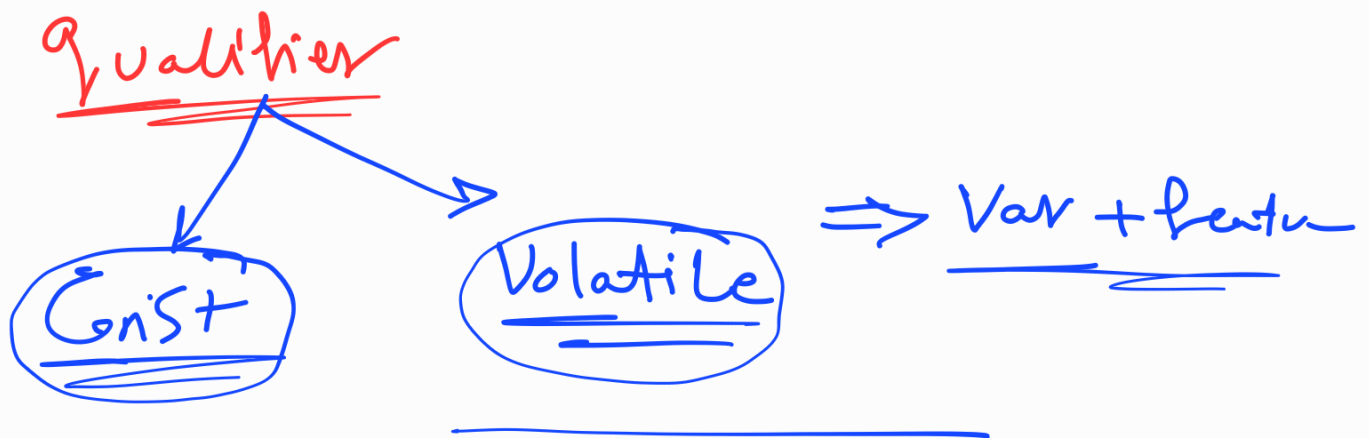


register int x;

⇒ (&x) ⇒ Compiler error

more than one static class

Static register int x; ⇒ error

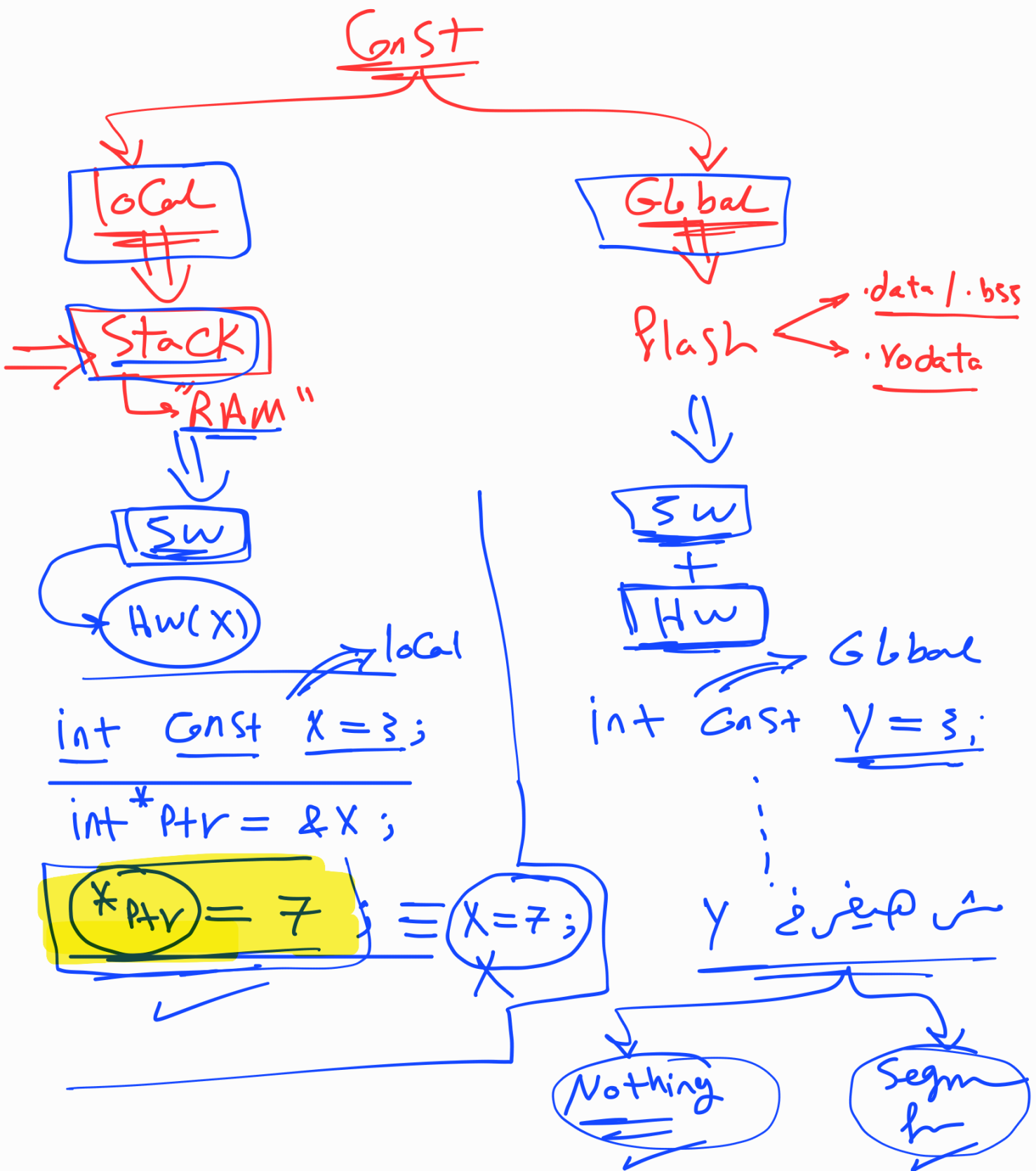


II Const

→ local & Global ⇒ Read only
(Const Value)

→ while def ⇒ Const int x = 10;

x = 10 ✗



⇒ Volatile

⇒ (Optimization) ⇒ Reduce

→ levels

→ #inst -

→ mem access time

→ Power Const -

00
01
02
03

~~00~~

- No opt —
- Not recm
- fastest → Compilation time.
- debuggin friendly
 - develop

(X)

- ~~0+~~
- 01 → decreases mem Access
 - 02 → Not debug fr
 - 03
 - Ruckl op —
 - Slow compIL time
 - 02 + Aggressive Step —
 - Not debug fr
 - Slowest comp-time
 - May Cause bugs in the Prog.

Volatile

