

Aide cours CM3 Allocation Dynamique

Exemple 2

Déclaration

```
typedef struct S_s {
    char c1 ;
    double c2 ;
} S_t ;
int * pint ;
S_t * pst ;
```

Initialisation

```
int v1 ;
S_t v2 ;
pint = &v1 ;
pst = &v2 ;
```

Utilisation

```
(* pint) = 99 ;
(* pst).c1 = 'a' ;
(* pst).c2 = 33333 ;
```

Simplification : $(*p).c \Leftrightarrow p->c$ Donc le listing peut s'écrire :

```
(* pint) = 99 ;
pst->c1 = ' a ' ;
pst->c2 = 33333 ;
```

Attention

```
v2.c1 = 'a' ; /* OK */ v2->c1 = 'a' ; /* KO */
pst.c1 = 'a' ; /* KO */ pst->c1 = 'a' ; /* OK */
```

Suppression

– On ne peut pas détruire (libérer la place mémoire prise par) un objet statique
– on peut en revanche détruire le pointeur sur celui ci (NULL)
pint = NULL ;
pst = NULL ;

Array and Pointers

```
/* an array with 5 elements */
double balance[5] = {1000.0, 2.0, 3.4, 17.0, 50.0};
double *p;
int i;
```

```
p = balance;
```

Right left thingy for complex declarations

I call it the Right-Left Thingy(TM)

Start at the variable, then go right, and left, and right...and so on.

```
int* arr1[8];
```

arr1 is an array of 8 pointers to integers.

```
int (*arr2)[8];
```

arr2 is a pointer (the parenthesis block the right-left) to an array of 8 integers.

```
int *(arr3[8]);
```

arr3 is an array of 8 pointers to integers.