

Programmation orientée objet

Rappel sur les symboles * et &

Déclaration de variables ou d'objets * et &

- Définition d'un pointeur:

```
int * Ptr = 0;
```

Lors de la définition d'un pointeur, on peut uniquement initialiser la valeur de ce pointeur (et non la valeur vers laquelle il pointe).

Par exemple:

float* ptr= nullptr valide

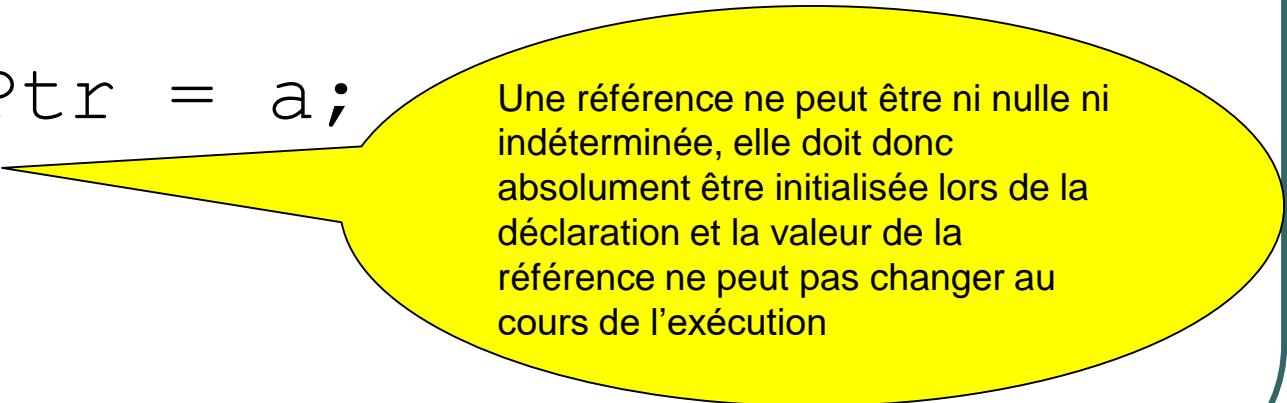
float* ptr= 10.0 non valide

Déclaration de variables ou d'objets &

- Définition d'une référence: permet de manipuler une variable sous un autre nom:

```
int a ;
```

```
int & Ptr = a;
```



Une référence ne peut être ni nulle ni indéterminée, elle doit donc absolument être initialisée lors de la déclaration et la valeur de la référence ne peut pas changer au cours de l'exécution

Section des instructions: opérateur *

Multiplication:

```
res = a * b;
```

Déréférencement d'un pointeur:

```
*Ptr = 10;
```

Section des instructions: opérateur &

- Opération Et binaire:

```
res = a & b;
```

- Adresse d'une variable:

Si on a les déclarations suivantes

```
int a ;
```

```
int * Ptr;
```

```
Ptr = &a;
```

Déclaration

Lorsqu'on déclare plusieurs pointeurs/références du même type, il faut répéter le */&.

Par exemple:

`int* a, * b, * c;` (déclare trois pointeurs)

`int* a, b, c;` (déclare un pointeur et 2 entiers)

`int* u, & v = b , w;` (déclare un pointeur, une référence et un entier)