

## Retour sur les références du C++

- Etant donné un **type T**, C++ offre au programmeur le **pseudo-type T &**
  - Référence sur un objet de type T
  - Une référence correspond à un synonyme ou alias

Ex :  

```
int a;
int &b=a; //b est un alias de a
        (référence sur la variable a)
```

LIF5 - 2004-2008 R. Chaîne

1

- La référence n'est **pas un vrai type**
- La définition d'une référence ne correspond pas à la définition d'une nouvelle variable
- Toute opération effectuée « sur » une référence est effectuée sur la variable référée (et inversement)

– Ex :  

```
int a=4;
int &b=a;
b++;
a++;
b=a;
```

LIF5 - 2004-2008 R. Chaîne

2

- Une référence **doit** être initialisée au moment de sa définition
    - Par une valeur modifiable (*-value*) dans le cas d'une référence **T &**
    - Par une valeur modifiable ou constante dans le cas d'une référence **const T &**
- Ex :
- ```
double d;
double& dr1=5; //Erreur
double& dr2=d; //OK
const double& cdr1=5; //OK
const double& cdr2=d; //OK mais pas de
                    //modif. de d via cdr2
```

LIF5 - 2004-2008 R. Chaîne


3

- Attention, la référence est associée à une variable au moment de sa définition, cette association ne peut pas être modifiée par la suite

LIF5 - 2004-2008 R. Chaîne

4


```
int a=3;
int &b=a;    b ≅ a
```



LIF5 - 2004-2008 R. Chaîne

5

```
int a=3;
int &b=a;    b ≅ a
```



- Secret de Polichinelle :  
Réalisation d'une référence à travers un «pointeur masqué»

pm\_b  
(pointeur masqué)



Chaque occurrence de b dans le programme est remplacée par \*pm\_b

LIF5 - 2004-2008 R. Chaîne

6

Rappel : Mise en œuvre du passage de paramètres **résultat** ou **donnée résultat** à une procédure en C++

```
void swap(int& a, int& b)
{ int c;
  c = a;
  a = b;
  b = c;
}
```

A l'appel  
int aa=4, bb=5;  
swap(aa,bb);

LIF5 - 2004-2008 R. Chaîne

7

Peut-on voir un appel de fonction à gauche d'un opérateur d'affectation?

LIF5 - 2004-2008 R. Chaîne

8

OUI!!!

S'il s'agit d'une fonction retournant une référence

```
Ex : int& elt(int tab[],int i,int t)
//précondition : tab de taille t (au moins)
//              : 0<=i<=t-1
//Résultat : retourne une référence
//          sur le ième élément de tab.
//          Attention cette référence pourra ensuite
//          permettre d'en modifier le contenu
{return tab[i];}
```

A l'appel :

```
int tatab[4];
elt(tatab,0,4)=2;
```

LIF5 - 2004-2008 R. Chaîne

9

Que penser de :

```
int& inc(int i)
{
  int temp=i+1;
  return temp;
}
```

LIF5 - 2004-2008 R. Chaîne

10