

COURS DE C++

10 - Héritage

RAFFINONS L'EXEMPLE PRÉCÉDENT

- On aimerait avoir deux types d'employés : les managers et les employés
- Un manager a un tableau d'employés sous ses ordres
- Un manager peut avoir un manager supérieur
- Un département contient un manager principal
- On aimerait donc avoir une classe Manager qui soit un sous-ensemble de la classe Employé

HÉRITAGE

- On dit alors que Manager hérite ou dérive de Employe
- On le définit ainsi :

```
class Manager : public Employe { };
```
- On doit définir des constructeurs propres mais on peut faire appel au constructeurs existants

```
Manager(string nom) : Employe(nom) { }
```
- Un objet de cette classe a accès à toutes les fonctions et méthodes de sa classe parente

PROBLÈME : APPEL DE MÉTHODE

- Par défaut `Manager m("a"); m.display();` va faire référence à la fonction `display` fournie par la classe `Employe`
- On aimerait pouvoir définir une variante pour les managers
- On peut redéfinir `display` dans la classe `Manager`
- On, dans la nouvelle fonction, faire référence à l'ancienne avec `Employe::display();`

PROBLÈME AVEC LES POINTEURS

- Si on fait `Employe *e = new Manager("toto");` que va faire `e->display();` ?
- On aimerait quand on a un pointeur prendre la meilleure fonction possible.
- Possible en déclarant dans `Employee` la méthode `display` comme étant virtuelle.