

Séance 4

Classe (3)

Création d'objet par new

Pointeur this

Fonction amie

Donnée statique

Fonction statique

Création d'objet par new

Possibilité de créer dynamiquement un objet par new.

Suppression par delete.

```
Date* p;  
p = new Date (2, 9, 2023);  
p->afficher();  
...  
delete p;
```

new crée l'objet et appelle le constructeur

delete appelle le destructeur (s'il existe) et supprime l'objet

Pointeur this

this :

- est un mot clé du langage, utilisable seulement à l'intérieur d'une fonction membre
- est un pointeur sur l'objet courant (mis à jour automatiquement par le compilateur)
- il permet d'exprimer
 - l'adresse de l'objet courant : `this`
 - une référence sur l'objet courant : `*this`

```
void Date::afficher() const {  
    // this est disponible et pointe sur l'objet courant  
    cout << jour << ...  
}
```

équivalent à `this->jour`

Fonction amie

Fonction amie d'une classe : fonction non membre, à qui la classe permet d'accéder à la partie privée.

```
class Date
{
public :
    Date(int, int, int);
    void afficher() const;
    // prototype précédé de friend
    friend bool premSemestre(const Date& da);
private :
    int jour, mois, annee;
};
```

ex : fonction non membre testant si une date est au 1^{er} semestre

```
bool premSemestre(const Date& da) {
    if (da.mois <= 6) return true;
    else return false;
}
```

accès à mois interdit (privé)
friend => accès à mois autorisé

```
Date d(...);
if (premSemestre(d) == true)
    ...
```

fonction non membre => pas d'objet courant,
tous les objets sont en paramètre

Eviter de déclarer des fonctions amies, écrire en priorité des fonctions membres.

Les fonctions amies sont utiles pour certains opérateurs dans les classes (voir séance sur les opérateurs).

Donnée statique

Donnée statique : donnée membre avec **un seul exemplaire** pour toute la classe.

Exemple : on veut compter le nb d'objets Date créés.

```
class Date
{
public :
    Date(int, int, int);
    ...
private :
    int jour, mois, annee;
    static int cpt;
};
```

```
Date::Date(int j, int m, int a) {
    jour = j;
    ...
    cpt++;
}
```

```
Date d1(...); => cpt++
Date d2(...); => cpt++
```

d1 contient : jour-mois-annee
d2 contient : jour-mois-annee
1 seul cpt

Important : une donnée statique **doit être initialisée** pour être créée en mémoire.

```
int Date::cpt = 0;      à placer dans Date.cpp avant l'écriture des fonctions
```

Fonction statique

Fonction statique : fonction membre qui ne s'applique pas sur un objet.

```
class Date
{
public :
    Date(int, int, int);
    ...
    static void afficherCpt();

private :
    int jour, mois, annee;
    static int cpt;
};
```

```
void Date::afficherCpt()
{
    cout << cpt << endl;
}
```

```
void main() {
    Date::afficherCpt();    affiche 0
    Date d1(...);
    Date d2(...);
    Date::afficherCpt();    affiche 2
}
```

fonction membre, mais pas d'objet courant, this non utilisable