

Constructeur

Valeurs par défaut

- La classe Employee peut avoir deux constructeurs:

```
class Employee
{
public:
    Employee(); // par défaut
    Employee(string name, double salary); // par paramètres
    ...
};
```

- Le constructeur par défaut met la chaîne “unknown” et la valeur 0 aux attributs nom_ et salaire_, respectivement

Valeurs par défaut (suite)

- Étant donné que le constructeur par paramètres prend les valeurs fournies et les copie dans les attributs, on se contente de celui-ci si on définit des valeurs par défaut aux paramètres dans seulement la définition de la classe:

```
class Employee
{
public:
    Employee(string name = "unknown",
              double salary = 0);
    ...
};
```

Valeurs par défaut (suite)

- En fait, c'est comme si nous avions maintenant trois constructeurs:

```
int main()
{
    Employee anonyme;
    Employee marie("Marie");
    Employee paul("Paulo", 45000);
    ...
}
```

Délégation Constructeur C++11

Délégation Constructeur

- En C++98, si on veut que 2 constructeurs fassent la même chose
 - répéter les instructions ou
 - créer une fonction membre init()
- En C++11, on peut définir un constructeur à l'aide d'un autre constructeur.

Délégation Constructeur

```
Employee::Employee(): Employee(200.00) {  
}  
  
Employee::Employee(string name) : Employee(200.00)  
{  
    name_ = name;  
}  
  
Employee::Employee(double salary)  
{  
    salary_ = salary;  
}
```



Permet
d'assigner un
salaire de base

Initialisation des attributs C++11

Définition d'une classe

En C++11

- Il est maintenant possible d'initialiser des attributs lors de la définition de la classe
- Lors de la construction de l'objet
 - l'attribut est construit en utilisant cette initialisation,
 - si pas indiqué autrement dans la liste d'initialisation du constructeur.
 - Initialisation faite avant d'exécuter le corps du constructeur.

Initialisation des attributs

```
class Employee
{
public:
    Employee();
    Employee(string name);
    Employee(string name, double salary);
    double getSalary() const;
    string getName() const;
    void setSalary(double salary);

private:
    // Attributs
    string name_ = "unknown";
    double salary_ = 0;
};
```

Initialisation des attributs

```
Employee::Employee()  
{  
}
```

name_ = "unknown"
salary_ = 0

```
Employee::Employee(string name)  
{  
    name_ = name;  
}
```

```
Employee::Employee(string name, double salary)  
{  
    name_ = name;  
    salary_ = salary;  
}
```

(Sera vu plus tard comment éviter la double initialisation)

Initialisation des attributs

- Il est également possible d'initialiser un attribut à partir d'une expression utilisant attributs, opérateurs (dont new), fonction (membre ou pas) :

```
class uneClasse {  
    private:  
        int x_ = rand() % 5 ;  
        int y_ = 2 * x_ ;  
        double* z_ = new double;  
};
```