

Année 2011-2012

**Cours de C++
aux CPI 2**

Fiche de présentation du C++

Le langage C++ :

- 1979 : C avec classes
- 1983 : C avec classes devient C++
- Concepteur : Bjarne Stroustrup
- Surcouche du langage C
- Trois paradigmes
 - Langage procédural
 - Langage orienté objet
 - Langage générique



C++ = C Ansi + POO + Améliorations du C

Librairie standard d'entrées/sorties de C++

- C++ a redéfini les fonctions d'entrées/sorties pour simplifier leur utilisation
- Il faut inclure le fichier de prototypage suivant :
`#include <iostream>`
- les variables globales `stdin`, `stdout` et `stderr` sont remplacées respectivement par `cin`, `cout` et `cerr`.
- l'opérateur `<<` remplace la fonction `printf`.
- l'opérateur `>>` remplace la fonction `scanf`.
- Cette librairie a été définie à l'intérieur de l'espace de noms `std`. Donc une fonction `f` s'appelle en réalité `std :: f`. Il faut utiliser la clause `using namespace std;` pour éviter de mettre le préfixe `std ::`.



Librairie standard d'entrées/sorties de C++ : Exemple

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    int a,b;  
  
    cout << "saisir a";  
    cin >> a;  
    cout << "le carré de " << a << " vaut " << a*a << endl;
```



Classe C++ : Introduction

- C++ est une couche supplémentaire du C. On parlera de fonctions indépendantes pour les fonctions C et de méthodes pour les fonctions d'une classe.
- L'extension des fichiers C++ est .h pour les fichiers d'entêtes (comme en C) et .cpp pour la définition du code des fonctions indépendantes et des méthodes
- En C++ une classe est définie dans deux fichiers :
 - un fichier header dans lequel on définit les types des attributs et les prototypes des méthodes;
 - un fichier .cpp pour définir le code des méthodes de la classe.

Classe C++ : Une classe très simple

Le fichier point.h

```
class Point {  
    private :  
        int ;  
        int y;  
  
    public :  
        void setX(int);  
        void setY(int);  
        int getX();  
        int getY();  
};
```

Le fichier point.cpp

```
#include "point.h"  
void Point :: setX(int px) {  
    x = px;  
}  
void Point :: setY(int py) {  
    y = py;  
}  
int Point :: getX() {  
    return x;  
}  
int Point :: getY() {  
    return y;  
}
```

ATTENTION : ne pas oublier le ;

Classe C++ : une classe très simple : Exemple

```
#include <iostream>
using namespace std;

int main() {
    int a,b;
    Point p;
    cout << "saisir x";
    cin >> a;
    cout << "saisir y";
    cin >> b;

    p.setX(x);
    p.setY(y)
    cout << " x = " << p.getX() << " y = " << p.getY();
}

```



Classe C++ : Une classe simple : Constructeur

Le fichier point.h

```
class Point {  
    private :  
    int x;  
    int y;  
  
    public :  
    Point (int px,int py);  
    int getX();  
    int getY();  
};
```

ATTENTION : ne pas oublier le ;

Le fichier point.cpp

```
#include "point.h"  
Point :: Point(int px,int py) {  
    x = px;  
    y = py;  
}  
int Point :: getX() {  
    return x;  
}  
int Point :: getY() {  
    return y;  
}
```

Classe C++ : une classe simple : Constructeur : Exemple

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    int a,b;  
    cout << "saisir x";  
    cin >> a;  
    cout << "saisir y";  
    cin >> b;  
  
    Point p(a,b);  
  
    cout << " x = " << p.getX() << " y = " << p.getY();  
}
```



Surcharge en C++

- La surcharge consiste à écrire deux fonctions qui ont le même nom mais pas le même prototypage.
- La surcharge n'est pas possible en C mais l'est en C++.
- On peut surcharger des méthodes comme des fonctions indépendantes
- Exemple :

```
int max(int x, int y);
```

```
float max(float x, float y);
```

- Remarque : un constructeur est une méthode (particulière) d'une classe qui a pour nom le nom de la classe. On peut donc avoir plusieurs constructeurs à condition qu'ils aient des prototypages différents.