Aide mémoire : du C au C++

Le langage C++ est basée sur la syntaxe du C. Leur différence principale consiste dans le fait que le C++ est un langage orienté objet Néanmoins, il existe quelques différences supplémentaires entre les langages C et C++ qui sont résumées dans ce document. Ces différences sont principalement dues au fait que le langage C++ est un langage plus haut niveau que le C et qu'il permet d'utiliser des librairies évoluées.
On peut compiler le C++ avec les compilateurs `cc`, `g++` ou `gpp` sous Unix/Linux ou avec `Visual C++` ou `C++ Builder` sous Windows.

1 Entrées/Sorties

`#include<iostream>` La bibliothèque `iostream.h` permet de gérer les flots d'Entrée-Sortie écran et clavier.

`cout` On écrit à l'écran (sortie standard) avec la commande `cout` qui s'utilise selon l'exemple suivant:

```
cout«"La variable x vaut "«x«" millilitres"«endl;
```

Cet exemple affiche `La variable x vaut 10 millilitres` puis saute une ligne (si la variable `x` contient 10). Vous pouvez remarquer qu'il est inutile de marquer le type de la variable `x`.

`cin` On lit une variable ou une chaîne de caractères au clavier par la commande `cin` selon l'exemple suivant:

```
cin»x;
```

Cette commande demande à l'utilisateur d'entrer une valeur dans la variable x. Vous pouvez remarquer qu'il est inutile de marquer le type de la variable `x`. (Attention, si `x` est une chaîne de caractères, il faut bien sûr qu'elle soit allouée.)

format Si l'on désire formater l'affichage d'un entier, on peut utiliser la commande `setw` de la bibliothèque `iomanip`. Par exemple, `cout«setw(3)«i;` permet d'afficher l'entier `i` sur 3 caractères.

`#include<fstream>` La bibliothèque `fstream.h` permet de manipuler des fichiers.

`ofstream-ifstream` Les types `ofstream` et `ifstream` permettent de définir des objets qui se construisent avec pour paramètre le nom d'un fichier:

```
ostream fic_entree("sortie.txt");
istream fic_sortie("entree.txt");
```

Les objets `fic_entree` et `fic_sortie` sont alors des objets du même ordre que, respectivement, `cout` et `cin`. On les manipule de la même façon. Ainsi, `fic_entree«i;` inscrit le contenu de la variable `i` dans le fichier sortie.txt et et `fic_sortie»x;` lit le fichier entree.txt et remplit la variable `x` avec son contenu. Après ces opérations, on referme le fichier avec les appels `fic_entree.close()` et `fic_sortie.close()`.

2 Références et fonctions

référence Une référence est un alias (un autre nom) que l'on peut donner à une même variable. Par exemple la séquence

```
int i;
int &ref_i=i;
```

définit la variable entière `i` et une référence `ref_i` sur cette même variable. Ainsi, le programme va se conduire comme si une variable ainsi définie possédait 2 noms. (Il s'agit en fait d'un habillage haut-niveau de pointeurs).

paramètre On peut donc en C++ faire passer des paramètres par référence (en plus du passage par valeur du C). Par exemple, la fonction

```
void met_au_cube(double &x){
    x=x*x*x*; }
```

retourne la variable `x` mise au cube. Dans la fonction, la variable `x` manipulée est très exactement la variable passée en paramètres.

3 Allocation mémoire

`new delete` Les commandes d'allocation mémoire du C++ sont `new` et `delete`. Elle s'utilise ainsi:

```
int *i;
i= new int;
delete i;
```

tableau Il existe une allocation particulière pour les tableaux en C++. On définit un tableau par la commande `T= new int[10];` et on le libère avec `delete[] T;`.

4 Quelques autres points

déclaration Une variable peut être déclarée n'importe où en C++, sa validité se limitant alors de sa déclaration jusqu'à la fin du bloc (zone entre accolade) où elle est définie. Ce n'est pas parce que c'est autorisé qu'il faut le faire, ceci étant assez dangereux. En revanche, cette opération est très utile lorsqu'il s'agit de définir une variable de flot Entrée/Sortie par exemple.

`bool` Le C++ dispose d'un type booléen, dont les valeurs sont 1 pour vraie et 0 pour faux.

`string` La librairie standard du C++ possède le type `string` (dans `string.h` qui correspond à une chaîne de caractère de haut-niveau. Elle n'a pas besoin d'être allouée et permet les manipulations de chaîne. En revanche sont comportement dépend fortement des compilateurs.

`exception` On peut gérer les anomalies par les exceptions. Une anomalie est une erreur qui ne peut pas (ou ne doit pas) être traitée localement (exemple: erreur de lecture fichier). On lève alors une exception qui pourra être par la suite réglée par un gestionnaire d'exception placé à part du programme.

surcharge Le C++ autorise la surcharge des fonctions. C'est-à-dire qu'un programme peut manipuler des fonctions homonyme (de même nom mais de liste de paramètres différents). Le C++ autorise aussi la surcharge d'opérateurs (par exemple, on peut définir la surcharge de l'opérateur `+` entre deux matrices).

généricité Le C++ autorise la généricité. C'est-à-dire que l'on peut coder une fonction ou un ensemble de variables et de fonctions en paramétrant le type de certains de ces composants. La généricité permet un code plus portable et surtout réutilisable pour différents types.

STL Le C++ possède, dans sa libraire standard, une librairie standard générique (STL) qui lui permet de manipuler des structures de données complexes comme les listes chaînées, les arbres,... Elle est basée sur les concepts de conteneurs (vector, list, set,...), d'itérateurs et d'algorithmes.

Objet Le C++ est néanmoins fondamentalement un langage de programmation orienté objet qui permet de concevoir du code objet avec tous ces concepts (objets, classes, abstraction, encapsulation, modularité, polymorphisme, héritage,...)