

Mémo C++

Vous trouverez ici tout ce que vous devez savoir pour coder vite fait en C++... Mais ne vous réjouissez pas trop vite, il faudra apprendre d'autres choses que ce qu'il y a dans ces 2 feuilles !

Il y a une chose à ne pas oublier quand on code en C++ : les **minuscules et les majuscules** sont importantes. Par exemple, la variable `x` et la variable `X` seront considérées comme étant deux variables différentes.

Ensuite, et j'insisterais souvent dessus, il faut mettre des **commentaires** pour expliquer ce que l'on veut faire. Un commentaire n'est pas interprété par le compilateur, il ne sert qu'au programmeur. Il y a deux manières de faire des commentaires en C++ :

```
//ceci est un commentaire sur une ligne
//je doit remettre les deux barres si j'ai besoin d'une autre ligne

/*ceci est un commentaire
Sur plusieurs lignes.
Pas besoin de remettre le signe ! */
```

Enfin, et ce sera la dernière chose que vous devrez savoir pour l'instant, c'est que chaque instruction se termine par **un point virgule**. Bien souvent, il faut mettre un point-virgule à la fin de chaque ligne. Cependant, il y a des cas où il ne faut pas : c'est notamment le cas pour les « `if` », « `while` » et d'autres structures que nous verrons plus tard. Le seul moyen de ne pas faire d'erreur est de se servir des exemples donnés à la page 3.

L'affichage

Afin de pouvoir afficher des trucs à l'écran, il faut utiliser une instruction un peu bizarre : `cout`. Elle fonctionne grâce à un opérateur : `<<`. Cet opérateur permet d'envoyer ce qu'on veut afficher dans l'opérateur. Ainsi, si on veut afficher « hello world », on écrira `cout<< « hello world »`. Cependant, `cout` ne gère pas le retour à la ligne. Ainsi, il faut rajouter le mot clef `endl` pour rajouter un retour à la ligne : `cout<< « hello world » <<endl`. On peut aussi mettre bout à bout des variables, des phrases, etc. Par exemple, on peut faire un truc du genre : `cout<< « valeur : » << var1<<endl`.

La saisie

La saisie s'effectue avec une instruction un peu pareil : `cin`. Sauf que là, c'est l'instruction qui envoie des données vers la variable, donc on utilisera l'opérateur `>>`. Ainsi, pour saisir une valeur dans la variable `age`, on écrira : `cin>>age`. De même que pour `cout`, on peut mettre bout à bout plusieurs variables : `cin>>nom>>prenom`.

Déclaration des variables

Contrairement à l'algorithmique, il est possible de déclarer les variables où on veut... Cependant, il y a deux règles à respecter :

- 1- La variable doit être déclarée avant son utilisation
- 2- La variable doit être déclarée dans le bloc (ou un bloc qui contient le bloc) où elle est utilisée.

Un bloc est délimité par des guillemets : {} (les guillemets sont utilisés par `if`, `while`, etc...).

Exemple :

```
{
    int A;
    {
        int B;
        //on a accès à A et B
    }
    int C;
    //on a accès à A et C mais pas B !
}
```

Comme on le voit sur l'exemple, la déclaration d'une variable se fait dans l'autre sens qu'en algorithmique : c'est le type suivi du nom de la variable.

Comparaison des opérateurs :

Opérateurs de calculs :

opérations possibles	Pseudo-code	C++
addition	+	+
soustraction	-	-
multiplication	*	*
Division réelle	/	/
Division entière	DIV	/
Modulo	MOD	%

Opérateurs de comparaison :

opérations possibles	Pseudo-code	C++
Égal	=	==
Différent	≠	!=
Supérieur	>	>
Supérieur ou égal	≥	≥
Inférieur	<	<
Inférieur ou égal	≤	≤

Opérateurs logiques :

opérations possibles	Pseudo-code	C++
Négation	NON	!
Conjonction	ET	&&
Disjonction	OU	(AltGr-6)

Liens entre le C++ et l'algorithmique

Algorithmique

Structure du programme :

Début
...
Fin

Déclaration des variables :

Toto : entier
Tata : chaîne

Type des variables :

Entier
Réel
Caractère
Chaîne
Booléen

Affectation :

X ← 2

Structure conditionnelle :

Si x < 2
Alors
...
Sinon
...
FinSi

Structure itérative :

TantQue x < 2
Faire
...
FinTantQue

Pour i de 1 à 10
Faire
...
FinPour

Liens pour les conditions :

ET
OU
NON

C++

Structure du programme :

```
Void main() {  
...  
}
```

Déclaration des variables :

```
int Toto;  
string Tata;
```

Type des variables :

```
int  
float OU double  
char  
string  
bool
```

Affectation :

```
X = 2
```

Structure conditionnelle :

```
if (x < 2)  
{  
...  
} else {  
...  
}
```

Structure itérative :

```
while (x < 2)  
{  
...  
}  
  
For(int i=1 ; i < 10 ; i++)  
{  
...  
}
```

Liens pour les conditions :

```
&&  
||  
!
```

A droite, vous avez les fichiers correspondant au code c++ correspondant aux algorithmes.

```
Programme puissance2
Var x : entier
Debut
Afficher « saisissez un chiffre : »
Saisir x
Afficher « le chiffre au carré est égal à : »,x*x
Fin
```

puissance2.cpp

```
#include <iostream>
#include <string>
using namespace std;
void main(){
int x;
cout<<« saisissez un chiffre : »<<endl;
cin>>x;
cout<< « le chiffre au carré est égal à : »<<x*x<<endl;
};
```

```
Programme majeur
Var age : entier
Debut
Afficher « saisissez votre age : »
Saisir age
Si age>18
Alors
Afficher « vous êtes majeur »
Sinon
Afficher « vous êtes mineur »
FinSI
Fin
```

majeur.cpp

```
#include <iostream>
#include <string>
using namespace std;
void main(){
int age;
cout<< « saisissez votre age : »<<endl;
cin>>age;
if(age>18)
{
cout<< « vous êtes majeur »<<endl;
}else{
cout<< « vous êtes mineur »<<endl;
};
};
```

```
Programme caddie
Var prix,total : entier
Debut
Afficher « saisissez le prix de l'article : »
Saisir prix
total<-0
tantQue prix>0 faire
total<-total+prix
Afficher « saisissez le prix de l'article : »
Saisir prix
FinTantQue
Afficher « le total vaut : »,total
Fin
```

caddie.cpp

```
#include <iostream>
#include <string>
using namespace std;
void main(){
int prix,total;
cout<< « saisissez le prix de l'article : »<<endl;
cin>>prix;
total=0;
while(prix>0)
{
total=total+prix;
cout<< « saisissez le prix de l'article : »<<endl;
cin>>prix;
};
cout<< « le total vaut : »<<total<<endl;
};
```