
Errata associé à la version « 1.01 » des notes de cours
Date du 13 mai 2007

N'hésitez pas à me signaler les erreurs s'il y'en a.

Chapitre -1- Généralités

Page -11-

Incorrecte	g++ toto.cpp -o toto.o
Correcte	g++ toto.cpp -o toto.exe

Même si l'instruction reste correcte, le choix de l'extension « -o » n'est pas approprié. Il fait plutôt penser à un fichier objet et non pas à un fichier exécutable.

Incorrecte	g++ -g toto.cpp -o toto.o
Correcte	g++ -g toto.cpp -o toto.exe

Pour les mêmes raisons.

Chapitre -2- Identificateurs, types de base et variables

Page -14-

Incorrecte	Le premier caractère doit être une lettre, un digit ou bien un souligné.
Correcte	Le premier caractère doit être une lettre ou bien un souligné.

Le premier caractère ne peut pas être un « digit ». L'exemple sur la même page le montre clairement.

Page -17-

Incorrecte	9. 9.0 9E1 Légales : « double » 3.0
Correcte	9. 9.0 9E1 Légales : « double » 9.0

Le chiffre « 9.0 » et non pas « 3.0 ».

Chapitre -3- Entrée/Sortie en C++

Page -28-

Incorrecte	
Correcte	Il manque le « using namespace » dans le programme.

```
#include <iostream>
#include <iomanip>
```

```
using namespace std ;
```

```
int main(){
// etc.
}
```

Chapitre -5- Opérateurs, expressions et conversion

Page -48-

Incorrecte	Op1 : 0 !Op1 : 0 (et) Op1 : 1 !Op1 : 1
Correcte	Op1 : 0 !Op1 : 1 (et) Op1 : 1 !Op1 : 0

Dans l'exemple de la page 48, l'opérateur de négation appliqué à « Op1 » ne donne pas les bons résultats ! La règle est comme suit : si la valeur de « Op1 » vaut « 0 », la négation « !Op1 » donne une valeur de « 1 ». Et si la valeur de « Op1 » vaut « 1 », la négation « !Op1 » donne la valeur « 0 ».

Chapitre -6- Les structures de contrôle

Page -63-

Incorrecte	if (x == 5) break ;
Correcte	if (i == 5) break ;

Le nom de la variable est « i » et non pas « x ».

Page -64-

Incorrecte	if (x == 5) break ;
Correcte	if (i == 5) break ;

Pour les mêmes raisons. Le nom de la variable est « i » et non pas « x ».

Chapitre -7- Les fonctions

Page -66-

Incorrecte	I n'y a pas de fonctions en Java ...
Correcte	Il n'y a pas de fonctions en Java ...

Page -75-

Incorrecte	double exemple(double x, char c= "T",double z=200.5)
Correcte	double exemple(double x, char c= 'T',double z=200.5)

Le 2^e argument de la fonction « exemple » est un caractère et non pas une chaîne de caractère.

Incorrecte	w = exemple(a, "s",-670.9) ;
Correcte	w = exemple(a,'s',-670.9) ;

Pour les mêmes raisons.

Incorrecte	w = exemple(a, "s") ;
Correcte	w = exemple(a,'s') ;

Pour les mêmes raisons.

Page -76-

Incorrecte	150.8 salut -670.9
Correcte	150.8 s -670.9

On passe comme 2^e argument la lettre « s » et non pas la chaîne « salut ». L'affichage en sortie est donc la lettre « s ».

Incorrecte	-670.9 150.8 salut
Correcte	-670.9 150.8 s

Pour les mêmes raisons.

Incorrecte	200.5 150.8 rien
Correcte	200.5 150.8 T

On passe comme 2^e argument la lettre « T » et non pas la chaîne « rien ». L'affichage en sortie est donc la lettre « T ».

Page -78-

Incorrecte	
Correcte	Il manque le « using namespace » dans le programme.

```
#include <iostream>
#include <iomanip>
```

```
#define MAX(a,b) (((a) > (b)) ? (a) : (b) )
```

using namespace std ;

```
int main(){
// etc.
}
```

Chapitre -8- Pointeurs & Allocation mémoire

Page -89-

Incorrecte	int a[2][4] = {{1,2,3,4},{5,6,7,8}} ;
Correcte	int tab [2][4] = {{1,2,3,4},{5,6,7,8}} ;

Le nom du tableau est « tab » et non pas « a ».

Incorrecte	int a[2][4] = {1,2,3,4,5,6,7,8} ;
Correcte	int tab [2][4] = {1,2,3,4,5,6,7,8} ;

Pour les mêmes raisons.

Page -91-

Incorrecte	un tableau de chaîne de caractères : 300 caractères
Correcte	un tableau de chaîne de caractères : 200 caractères

Le tableau « tab » est de taille « [10][20] » i.e. 10x20=200 caractères.

Incorrecte	Si chaque caractère occupe 1 octet en mémoire, cette définition réserve 300 octets en mémoire.
Correcte	Si chaque caractère occupe 1 octet en mémoire, cette définition réserve 200 octets en mémoire.

Pour les mêmes raisons.