

## TP 3 : Variables et constantes

### 1. La déclaration des variables

Les types de bases utilisés dans le langage C++ sont : le "*int*" (pour les entiers), le "*float*" (pour les réels), "*bool*" (pour représenter une variable booléenne) et le "*char*" (pour les caractères).

#### Syntaxe :

➤ Pour déclarer une variable en C++, on utilise la syntaxe suivante :

*Type Nom-variable1, Nom-variable2, .... ;*

#### Exemple :

*int A, C;*

*char Ch;*

*string Prénom;*

*float tx1, x2;*

*bool B ;*

➤ En même temps qu'on déclare une variable, il est possible de lui attribuer une valeur initiale.

#### Exemple :

*int a=34 ;    ou int a (34) ;*

*float x=10.5 ;    ou    float x (10.5) ;*

#### 1.1. Le type entier (int) :

➤ Le langage C++ possède plusieurs types de base pour désigner un entier.

- ✓ **int** : contient un entier de taille normale, positif ou négatif.
- ✓ **short int** : contient un entier de petite taille, positif ou négatif.
- ✓ **long int** : contient un entier de grande taille (32 bits), positif ou négatif.
- ✓ **long long int** : contient un entier de plus grande taille (64 bits), positif ou négatif.
- ✓ **unsigned int** : contient un entier de taille normale, positif ou nul.
- ✓ **unsigned short int** : contient un entier de petite taille, positif ou nul.
- ✓ **unsigned long int** : contient un entier de grande taille (32 bits), positif ou nul.

✓ **unsigned long long int** : contient un entier de plus grande taille (64 bits), positif ou nul.

#### 1.2. Le type réel

➤ Pour représenter un réel, il existe 3 types de base :

- ✓ **float** (simple précision)
- ✓ **double** (double précision)
- ✓ **long double** (précision étendue)

#### 1.3. Le type caractère (char)

➤ pour représenter un caractère, ce caractère peut être un chiffre, exemple '3', ou une lettre, exemple 's', ou un caractère spécial, exemple '?'.  
'3', ou une lettre, exemple 's', ou un caractère spécial, exemple '?'.

#### 1.4. Le type booléen (bool)

➤ Ce type est tout naturellement formé de deux valeurs notées **true** et **false**.

#### Remarque :

Les chaînes de caractères (*string*), n'est pas un type élémentaire (simple). Pour l'utiliser, il faut placer l'entête du fichier : **#include <string>**

## 2. Affichage (Ecriture) sur écran

L'affichage à l'écran de la valeur d'une expression

#### Syntaxe :

**cout << expression ;**

➤ Plus généralement on peut écrire :

**Cout << expression1 << expression2 <<.... ;**

#### Exemples :

➤ On peut afficher plusieurs variables en même temps comme suit :

**cout <<A<<x1<<Ch<< nom;**

➤ on peut aussi afficher un message en même temps qu'une variable comme suit :

```
cout<< "L'étudiant" <<nom<< "a une moyenne de " <<moymoy<< " sur 20";
```



Mohammed : c'est la valeur de la variable **nom**

15 : c'est la valeur de la variable **moy**

### 3. Lecture d'une variable

#### Syntaxe :

```
cin>>Nom-variable ;
```

Cette instruction lit (au clavier) une valeur et l'affecte à la variable.

➤ Plus généralement on peut écrire :

```
cin>> var1 >> ... >>varn ;
```

#### Exemples :

```
Cin>> A ;
```

```
Cin>> x1>>ch>>nom ;
```

#### TP 1:

Écrire 3 programmes : le premier lit et affiche un réel, le deuxième lit et affiche un caractère et le troisième lit et affiche une chaîne de caractères.

#### Remarque :

La fonction `getchar()` ; interrompt l'exécution du programme jusqu'à ce que l'utilisateur appuie sur une touche du clavier.

Supprimez la fonction `getchar()`; et exécuter le programme à nouveau ?

### 4. Les constantes

Les constantes ne changent pas de valeurs durant l'exécution du programme contrairement aux variables. Pour définir une constante dans le langage C++ on utilise le mot clé **const** ou le mot clé **#define**.

#### Syntaxe :

**Const Type Nom-constante = valeur ;**

#### Exemples:

```
const int A=5;
```

```
const float Pi=3.14;
```

```
const char Ch='a';
```

#### Remarque :

- On ne peut pas lire une constante avec la fonction cin>> on lui affecter une nouvelle valeur puisque la valeur de la constante ne change pas durant l'exécution du programme.

- Les constantes sont affichées de la même façon que les variables.

#### 10. les commentaires:

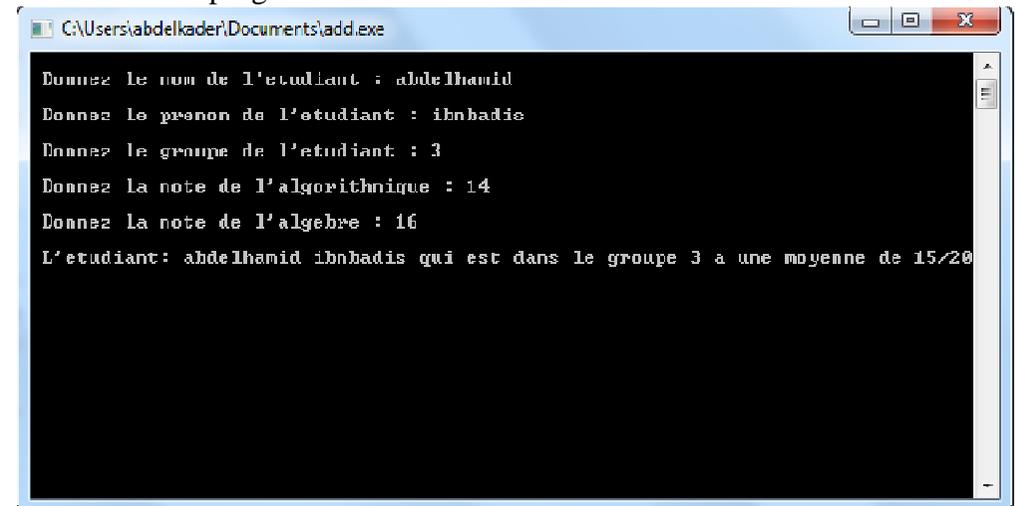
On peut ajouter des commentaires dans le code source d'un programmes en utilisant le symbole `/* */` ou le symbole `//` :

- On Commence par slash / suivie d'une étoile \* et on termine par une étoile \* suivie d'un slash : `/* plusieurs lignes */`
- On Commence par double slash pour un commentaire qui s'étale sur une seule ligne : `// Une seule ligne`

#### TP 2:

Écrire un programme C qui demande à l'utilisateur le nom, le prénom, le groupe, la note de l'algorithmique et la note de l'algèbre d'un étudiant puis affiche la moyenne. Utilisez les commentaires et les messages.

L'exécution du programme est comme suit :



## TP 3 (partie2): Les bibliothèques C++

### 1. les bibliothèques :

Une *bibliothèque* ou *librairie* logicielle est un ensemble de fonctions et procédures, regroupées et mises à disposition (dans un fichier) afin de pouvoir être utilisées directement sans avoir à les réécrire. Les fonctions et procédures sont regroupées de par leur appartenance à un même domaine conceptuel (entrées/sorties, mathématique, graphique, tris, ...).

Parmi les bibliothèques C++ standards on peut citer : **iostream**, **string**, **cmath**... .

- Pour utiliser une bibliothèque on ajoute au début du programme la ligne :  
**#include <nom\_de\_la\_bibliothèque>**

#### Exemple:

```
#include <iostream> // bibliothèque pour les instructions de lecture / écriture.
#include <string> // bibliothèque pour manipuler les chaîne de caractère.
#include <cmath> // bibliothèque pour les fonctions mathématiques.
```

·  
·  
·

#### Remarque :

L'instruction **#include <iostream>**, n'est pas suffisante pour utiliser les fonctions **cin** et **cout** ; ces fonctions sont défini dans l'espace de nom **std**, donc il faut rajouter l'instruction **using namespace std** ;

### 2. les opérateurs (opérations) en langage C++:

Les opérateurs sont des symboles permettant d'effectuer des opérations sur des valeurs. Parmi les opérateurs C++, On peut citer : ( ) - ++ -- % / \* - + .

Le tableau suivant donne la traduction en langage C++ des opérations les plus fréquentes en algorithmique.

Langage algorithmique	Langage C++
(A)	(A)
-A	-A
A ← A+1 (incréméntation)	A++, ++A A
A ← A-1 (décréméntation)	A--, --A
A mod B (modulo)	A % B
A / B (A reel ou B reel)	A / B
A / B (A et B entiers)	A / B
A * B	A * B
A + B	A + B
A - B	A - B
A > B	A > B
A ≥ B	A >= B
A < B	A < B
A ≤ B	A <= B
A = B	A == B
A ≠ B	A != B
A et B	A && B
A ou B	A    B
not(A)	!A
A ← B (l'affectation)	A = B
A ← A + B	A += B
A ← A - B	A -= B
A ← A * B	A *= B
A ← A / B	A /= B
A ← A mod B	A %= B

### 3. Priorités des opérateurs :

La priorité des opérateurs dépend du langage de programmation utilisé. Les opérateurs dans le langage C++ sont ordonnés comme suit :

Priorité	Opérateurs
1 (plus forte)	()
2	! - ++ --
3	% / *
4	- +
5	> >= < <=
6	== !=
7	&&
8	
9 (plus faible)	= += -= *= /= %=

- Un opérande entre deux opérateurs est associé avec l'opérateur qui a la plus forte priorité

#### Exemple :

$A + B * C \rightarrow (A + (B * C))$

- Un opérande entre deux opérateurs de même priorité est associé avec l'opérateur le plus à **gauche** sauf pour les opérandes ( ! - ++ -- = += -= \*= /= %= ) où il est associé avec l'opérateur le plus à **droite**.

#### Exemple :

$A * B / C \leftrightarrow ((A * B) / C)$   
 $A \% = B * = C \leftrightarrow (A \% = (B * = C))$

#### Remarque :

Pour éviter les erreurs d'association des opérateurs il est recommandé d'utiliser les parenthèses.

### 4. Quelques fonctions mathématiques

Les fonctions mathématiques sont définies dans la bibliothèque **cmath**, donc pour utiliser ces fonctions il faut d'abord inclure la bibliothèque au début du programme comme suit :

`#include <cmath>`

Parmi les fonctions contenues dans le fichier <cmath >, on peut citer :

Fonctions mathématiques	Langage C++
Puissance : $x^y$	<b>pow(x,y)</b> x de type float ou double
racine carree : $\sqrt{x}$	<b>sqrt(x)</b> x doit être positif
sinus(x)	<b>sin(x)</b>
Cosinus(x)	<b>cos(x)</b>
tangente(x)	<b>tan(x)</b>
logarithme népérien : $\log(x)$	<b>log(x)</b>
logarithme a base 10 : $\log_{10}(x)$	<b>log10(x)</b>
Exponentiel : $e^x$	<b>exp(x)</b>
La valeur absolue : $ x $	<b>abs(x)</b> s'applique sur les entiers

#### Remarque :

Attention au certains fonctions qui ne s'appliquent que sur un type particulier, elles peuvent faire des conversions implicites de type de leurs opérandes.

**Exemple :** `abs(-3.25)` donne **3**

#### TP :

- 1) Ecrire un programme C++ qui calcule les fonctions suivantes (utilisez les commentaires pour clarifier vos programmes):

$$f(x) = \frac{2(x+1)^2 - x}{x+1}$$

$$g(x) = (\sin(x) + \sqrt{x})^4$$

- L'exécution du programme est comme suit :

