

Langage C

Apprendre à lire...

François Cuvelier

Laboratoire d'Analyse Géométrie et Applications
Institut Galilée
Université Paris XIII.

26 septembre 2012

Plan

- 1 Les opérateurs
 - **Priorité des opérateurs**
 - Associativité des opérateurs
- 2 Evaluation des expressions
 - Ordre d'évaluation
 - Effets de bord
- 3 Lecture d'une déclaration
 - Méthodologie
 - Exemple 1
 - Exemple 2
 - Exemple 3
 - Exercices

tableau des priorités des opérateurs

Priorité	Type	Opérateurs	Associativité
0	Primaire	() [] . ->	G à D
1	Unaire	! ~ ++ -- + - & * (type) sizeof	D à G
2	Multiplicatif	* / %	G à D
3	Additif	+ -	G à D
4	Décalage	<< >>	G à D
5	Relationnel	< > >= <=	G à D
6	Egalité	== !=	G à D
7	Bits	&	G à D
8	Bits	^	G à D
9	Bits		G à D
10	Logique	&&	G à D
11	Logique		G à D
12	Conditionnel	? :	D à G
13	Affectation	= += -= ...	D à G
14	Virgule	,	G à D

Plus le numéro de priorité est faible, plus la priorité est forte.

Priorité des opérateurs

\triangle et \odot deux opérateurs **binaires**.

Propriété

Si l'opérateur \triangle a une priorité **plus forte** que l'opérateur ∇ alors

$$a \triangle b \nabla c \iff (a \triangle b) \nabla c$$

et

$$a \nabla b \triangle c \iff a \nabla (b \triangle c)$$

Exemple

Parenthéser l'instruction suivante :

$$x = 5 + 2 * y ;$$

► Identifier les opérateurs

Priorité des opérateurs

\triangle et \odot deux opérateurs **binaires**.

Propriété

Si l'opérateur \triangle a une priorité **plus forte** que l'opérateur ∇ alors

$$a \triangle b \nabla c \iff (a \triangle b) \nabla c$$

et

$$a \nabla b \triangle c \iff a \nabla (b \triangle c)$$

Exemple

Parenthéser l'instruction suivante :

$$x = 5 + 2 * y ;$$

► Déterminer les priorités des opérateurs

Priorité des opérateurs

\triangle et \odot deux opérateurs **binaires**.

Propriété

Si l'opérateur \triangle a une priorité **plus forte** que l'opérateur ∇ alors

$$a \triangle b \nabla c \iff (a \triangle b) \nabla c$$

et

$$a \nabla b \triangle c \iff a \nabla (b \triangle c)$$

Exemple

Parenthéser l'instruction suivante :

$$x = 5 + 2 * y ;$$

13 3 2

► Mettre les parenthèses en partant de l'opérateur de plus forte priorité

Priorité des opérateurs

\triangle et \odot deux opérateurs **binaires**.

Propriété

Si l'opérateur \triangle a une priorité **plus forte** que l'opérateur ∇ alors

$$a \triangle b \nabla c \iff (a \triangle b) \nabla c$$

et

$$a \nabla b \triangle c \iff a \nabla (b \triangle c)$$

Exemple

Parenthéser l'instruction suivante :

$$x = 5 \underset{13}{+} (2 \underset{2}{*} y);$$

► Mettre les parenthèses en partant de l'opérateur de plus forte priorité

Priorité des opérateurs

\triangle et \odot deux opérateurs **binaires**.

Propriété

Si l'opérateur \triangle a une priorité **plus forte** que l'opérateur ∇ alors

$$a \triangle b \nabla c \iff (a \triangle b) \nabla c$$

et

$$a \nabla b \triangle c \iff a \nabla (b \triangle c)$$

Exemple

Parenthéser l'instruction suivante :

$$x = (5 + (2 * y));$$

*(Note: In the original image, the number 13 is written below the 5, the number 3 is written below the +, and the number 2 is written below the *.)*

► Mettre les parenthèses en partant de l'opérateur de plus forte priorité

Exercice

Exemple

Parenthéser l'instruction suivante :

```
a = b > 1 && c < 0 ;
```

- Identifier les opérateurs

Exercice

Exemple

Parenthéser l'instruction suivante :

```
a = b > 1 && c < 0 ;
```

- ▶ Déterminer les priorités des opérateurs

Exercice

Exemple

Parenthésier l'instruction suivante :

$a = b > 1 \ \&\& \ c < 0 ;$
13 5 10 5

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité

Exercice

Exemple

Parthéniser l'instruction suivante :

$a = (b > 1) \&\& (c < 0);$

(Note: In the original image, the numbers 13, 5, 10, and 5 are written below the operators =, >, &&, and < respectively.)

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité

Exercice

Exemple

Parthéniser l'instruction suivante :

$a = ((b > 1) \&\& (c < 0));$

(Note: In the original image, the numbers 13, 5, 10, and 5 are written below the operators =, >, &&, and < respectively.)

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérandes)

- Les opérateurs \triangle et ∇ sont **associatifs de gauche à droite** :

$$a \triangle b \nabla c \iff (a \triangle b) \nabla c$$

- Les opérateurs \triangle et ∇ sont **associatifs droite à gauche** :

$$a \triangle b \nabla c \iff a \triangle (b \nabla c)$$

Exemple (1)

Parenthéser l'instruction suivante :

```
x = y += z = 3 ;
```

- Identifier les opérateurs

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérandes)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (1)

Parenthésier l'instruction suivante :

`x = y += z = 3 ;`

- Déterminer les priorités des opérateurs

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérandes)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (1)

Parenthéser l'instruction suivante :

$$x \underset{13}{=} y \underset{13}{+=} z \underset{13}{=} 3 ;$$

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérands)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (1)

Parenthéser l'instruction suivante :

$$x \underset{13}{=} y \underset{13}{+=} (z \underset{13}{=} 3);$$

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérands)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (1)

Parenthéser l'instruction suivante :

$$x \underset{13}{=} (y \underset{13}{+=} (z \underset{13}{=} 3));$$

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérands)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (2)

Parenthéser l'instruction suivante :

$$x = 3 * 5 / 2 ;$$

- Identifier les opérateurs

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérandes)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (2)

Parenthésier l'instruction suivante :

$$x = 3 * 5 / 2 ;$$

- ▶ Déterminer les priorités des opérateurs

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérandes)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (2)

Parenthéser l'instruction suivante :

$$x = 3 \underset{13}{*} 5 \underset{2}{/} 2 ;$$

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérands)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (2)

Parenthéser l'instruction suivante :

$$x = (3 \underset{13}{*} 5) \underset{2}{/} 2 ;$$

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Associativité d'opérateurs de même priorité

Propriété des opérateurs binaires (deux opérands)

- Les opérateurs \triangleleft et \triangleright sont **associatifs de gauche à droite** :

$$a \triangleleft b \triangleright c \iff (a \triangleleft b) \triangleright c$$

- Les opérateurs \triangleleft et \triangleright sont **associatifs droite à gauche** :

$$a \triangleleft b \triangleright c \iff a \triangleleft (b \triangleright c)$$

Exemple (2)

Parenthéser l'instruction suivante :

$$x = ((3 * 5) / 2) ;$$

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Associativité d'opérateurs de même priorité

Propriété des opérateurs unaires (une opérande)

Les opérateurs \odot et \otimes sont **associatifs de droite à gauche** :

$$\odot b \otimes \iff \odot (b \otimes)$$

Exemple

Parenthèser l'instruction suivante :

`x = - y ++ ;`

► Identifier les opérateurs

Associativité d'opérateurs de même priorité

Propriété des opérateurs unaires (une opérande)

Les opérateurs \odot et \otimes sont **associatifs de droite à gauche** :

$$\odot b \otimes \iff \odot (b \otimes)$$

Exemple

Parenthèser l'instruction suivante :

`x = - y ++ ;`

- ▶ Déterminer les priorités des opérateurs

Associativité d'opérateurs de même priorité

Propriété des opérateurs unaires (une opérande)

Les opérateurs \odot et \otimes sont **associatifs de droite à gauche** :

$$\odot b \otimes \iff \odot (b \otimes)$$

Exemple

Parenthésier l'instruction suivante :

$$x \underset{13}{=} \underset{1}{-} y \underset{1}{++} ;$$

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Associativité d'opérateurs de même priorité

Propriété des opérateurs unaires (une opérande)

Les opérateurs \odot et \otimes sont **associatifs de droite à gauche** :

$$\odot b \otimes \iff \odot (b \otimes)$$

Exemple

Parenthèser l'instruction suivante :

$$x \underset{13}{=} \underset{1}{-} (y \underset{1}{++});$$

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Associativité d'opérateurs de même priorité

Propriété des opérateurs unaires (une opérande)

Les opérateurs \odot et \otimes sont **associatifs de droite à gauche** :

$$\odot b \otimes \iff \odot (b \otimes)$$

Exemple

Parenthésier l'instruction suivante :

$$x = (- (y ++));$$

(Note: In the original image, the number 13 is written below the minus sign, and the number 1 is written below the first and last plus signs.)

- ▶ Mettre les parenthèses en partant de l'opérateur de plus forte priorité et en utilisant les règles d'associativité

Plan

- 1 Les opérateurs
 - Priorité des opérateurs
 - Associativité des opérateurs
- 2 Evaluation des expressions
 - **Ordre d'évaluation**
 - Effets de bord
- 3 Lecture d'une déclaration
 - Méthodologie
 - Exemple 1
 - Exemple 2
 - Exemple 3
 - Exercices

Ordre d'évaluation

A retenir

Pour les quatre opérateurs `&&`, `||`, `?:`, et `,` l'opérande le plus à gauche sera évalué en premier. L'ordre d'évaluation des opérandes pour tous les autres opérateurs n'est pas spécifié par le langage C.

Exemple

Pour évaluer l'expression $(3+2)(5-2)$, il faut calculer les deux opérandes de l'opérateur `*` : $(3+2)$ et $(5-2)$. Suivant le contexte (compilateur, système, ...), l'une ou l'autre de ces opérandes sera évaluée en premier !*

Ordre d'évaluation

A retenir

Pour les quatre opérateurs `&&`, `||`, `?:`, et `,` l'opérande le plus à gauche sera évalué en premier. L'ordre d'évaluation des opérandes pour tous les autres opérateurs n'est pas spécifié par le langage C.

Exemple

Pour évaluer l'expression $(3+2)(5-2)$, il faut calculer les deux opérandes de l'opérateur `*` : $(3+2)$ et $(5-2)$. Suivant le contexte (compilateur, système, ...), l'une ou l'autre de ces opérandes sera évaluée en premier !*

Mais est-ce gênant ?

Effets de bord

Définition (Effet de bord)

Un effet de bord consiste à modifier la valeur d'un opérande pendant le traitement d'une expression.

Remarque

Les opérateurs d'affectation, d'incrément et de décrémentation sont à effets de bord.

Effets de bord - Exemples

Exemple (1)

Etudier les instructions suivantes

```
1 | int x=0, a=0;  
2 | x=(a=2)+a;
```

$$x=(a=2)+a; \iff x=((a=2)+(a));$$

Suivant le *contexte*, l'opérande à gauche de l'opérateur `+` peut être évalué en premier ou en dernier.

- en premier, `x=4` et `a=2`.
- en dernier, `x=2` et `a=2`.

Effets de bord - Exemples

Exemple (2)

Etudier les instructions suivantes

```
1 | int x=0, a=0;  
2 | x = power(2, a=2) * power(3, a);
```

$x = \text{power}(2, a=2) * \text{power}(3, a); \iff x = (\text{power}(2, a=2)) * (\text{power}(3, a)) ;$

Suivant le *contexte*, l'opérande à gauche de l'opérateur `*` peut être évalué en premier ou en dernier.

- en premier, $x = \text{power}(2, 2) + \text{power}(3, 2) = 4 + 8 = 12$ et $a = 2$.
- en dernier, $x = \text{power}(2, 2) + \text{power}(3, 0) = 4 + 0 = 4$ et $a = 2$.

Effets de bord - Exemples

Exemple (3)

Etudier les instructions suivantes

```
1 | int n=0;  
2 | printf ("%d_%d", ++n, n+1));
```

L'ordre d'évaluation des arguments d'une fonction n'est pas spécifiée !

- si `++n` évalué avant `n+1`, on affiche 1 2,
- si `n+1` évalué avant `++n`, on affiche 1 1,

Effets de bord - Exemples

Exemple (4)

Etudier les instructions suivantes

```
1 | int i=0; a[3]={1,2,3};  
2 | a[i]=i++;
```

L'incrémentation a lieu après l'évaluation de l'expression à droite de l'opérateur d'affectation mais la norme ne précise pas l'ordre d'évaluation des opérandes de par et autre de l'opérateur d'affectation. L'incrément est effective avant l'exécution de l'instruction suivante.

1er cas	2ème cas	3ème cas
évaluation a[i]	évaluation i	évaluation i
évaluation i	évaluation a[i]	incrément i
incrément i	incrément i	évaluation a[i]
a={0,2,3} i=1	a={0,2,3} i=1	a={1,0,3} i=1

Plan

- 1 Les opérateurs
 - Priorité des opérateurs
 - Associativité des opérateurs
- 2 Evaluation des expressions
 - Ordre d'évaluation
 - Effets de bord
- 3 Lecture d'une déclaration
 - Méthodologie
 - Exemple 1
 - Exemple 2
 - Exemple 3
 - Exercices

A retenir

En partant du nom de l'objet déclaré, la lecture se fait de gauche à droite sauf si, parmi les constructeurs non encore lus, celui de gauche est prioritaire à celui de droite. Les parenthèses permettent de forcer la priorité.

Exemple (1)

```
int *f1( );
```

Exemple (1)

```
int *f1( );
```

- `f1` est le nom de l'objet déclaré.

Exemple (1)

```
int *f1 ( );
```

- `f1` est le nom de l'objet déclaré.
- `f1 ()` poursuite de la lecture de la gauche vers la droite car `()` est prioritaire sur `*` et `()` se lit *fonction* :

f1 est une fonction sans argument.

Exemple (1)

```
int *f1 ( );
```

- **f1** est le nom de l'objet déclaré.
- **f1 ()** poursuite de la lecture de la gauche vers la droite car **()** est prioritaire sur ***** et **()** se lit *fonction* :

f1 est une fonction sans argument.

- **f1 ();** fin de la déclaration à droite mais il reste des choses à lire à gauche.

Exemple (1)

```
int *f1 ( );
```

- **f1** est le nom de l'objet déclaré.
- **f1 ()** poursuite de la lecture de la gauche vers la droite car **()** est prioritaire sur ***** et **()** se lit *fonction* :

f1 est une fonction sans argument.

- **f1 () ;** fin de la déclaration à droite mais il reste des choses à lire à gauche.
- ***f1 () ;** ***** se lit *pointeur* :

f1 est une fonction sans argument retournant un pointeur.

Exemple (1)

```
int *f1 ( );
```

- **f1** est le nom de l'objet déclaré.
- **f1 ()** poursuite de la lecture de la gauche vers la droite car **()** est prioritaire sur ***** et **()** se lit *fonction* :

f1 est une fonction sans argument.

- **f1 () ;** fin de la déclaration à droite mais il reste des choses à lire à gauche.
- ***f1 () ;** ***** se lit *pointeur* :

f1 est une fonction sans argument retournant un pointeur.

- **int *f1 () ;**

f1 est une fonction sans argument retournant un pointeur de **int.**

Exemple

```
double *(*f2) (int);
```

Exemple

```
double *(*f2) (int);
```

- `f2` est le nom de l'objet déclaré.

Exemple

```
double * (*f2) (int);
```

- **f2** est le nom de l'objet déclaré.
- **(*f2)** à lire en premier car la priorité est forcée par les parenthèses :

f2 est un pointeur .

Exemple

```
double * (*f2) (int);
```

- **f2** est le nom de l'objet déclaré.
- **(*f2)** à lire en premier car la priorité est forcée par les parenthèses :

f2 est un pointeur .

- **(*f2) (...)** poursuite de la lecture de la gauche vers la droite car **()** est prioritaire sur ***** :

f2 est un pointeur de fonction.

Exemple

```
double *(*f2) (int);
```

- `(*f2) (int)` on lit l'intérieur des parenthèses qui décrit les arguments de la fonction :

f2 est un pointeur de fonction d'argument un int.

Exemple

```
double *(*f2) (int);
```

- **(*f2) (int)** on lit l'intérieur des parenthèses qui décrit les arguments de la fonction :

f2 est un pointeur de fonction d'argument un int.

- **(*f2) (int);** fin de la déclaration à droite mais il reste des choses à lire à gauche.

Exemple

```
double *(*f2) (int);
```

- `*(*f2) (int);` :

`f2` est un pointeur de fonction d'argument un `int` retournant un pointeur.

Exemple

```
double *(*f2) (int);
```

- `*(*f2) (int);` :

`f2` est un pointeur de fonction d'argument un `int` retournant un pointeur.

- `double *(*f2) (int);` :

`f2` est un pointeur de fonction d'argument un `int` retournant un pointeur de `double`

Exemple

```
int *(*f3[]) ();
```

Exemple

```
int *(*f3[]) ();
```

- **f3** est le nom de l'objet déclaré.

Exemple

```
int *(*f3[]) ();
```

- **f3** est le nom de l'objet déclaré.
- **f3[]** à lire en premier car **[]** est prioritaire sur *****. **[]** se lit *tableau* :

f3 est un tableau .

Exemple

```
int * (*f3[]) ();
```

- **f3** est le nom de l'objet déclaré.
- **f3[]** à lire en premier car **[]** est prioritaire sur *****. **[]** se lit *tableau* :

f3 est un tableau .

- **(*f3[])** à lire ensuite car la priorité est forcée par les parenthèses :

f3 est un tableau de pointeur.

Exemple

```
int * (*f3[]) ();
```

- `(*f3[]) ()` poursuite de la lecture de la gauche vers la droite car `()` est prioritaire sur `*` :

f3 est un tableau de pointeur de fonction.

Exemple

```
int * (*f3[]) ();
```

- `(*f3[]) ()` poursuite de la lecture de la gauche vers la droite car `()` est prioritaire sur `*` :

f3 est un tableau de pointeur de fonction.

- `(*f3[]) ();` fin de la déclaration à droite mais il reste des choses à lire à gauche.

Exemple

```
int *(*f3[]) ();
```

- `*(*f3[]) ();` :

f3 est un tableau de pointeur de fonction retournant un pointeur.

Exemple

```
int *(*f3[]) ();
```

- `*(*f3[]) ();` :

`f3` est un tableau de pointeur de fonction retournant un pointeur.

- `int *(*f3[]) ();` :

`f3` est un tableau de pointeur de fonction retournant un pointeur de `int`.

Exercice

Lire les déclarations suivantes :

- 1 `int *x1(double, int);`
- 2 `int * x2 [] [];`
- 3 `int ** x3 [];`
- 4 `int (*x4)(int *, int []);`
- 5 `int (*(* x5[3])(void))(char *);`

Correction

① `int *x1(double, int);`

`x1` est une fonction d'arguments un `double` et un `int`, retournant un pointeur de `int`

Correction

② `int * x2 [] [];`

x2 est un tableau de tableau de pointeur de `int`

Correction

③ `int ** x3 [];`

x3 est un tableau de pointeur de pointeur de `int`

Correction

④ `int (*x4)(int *,int []);`

x4 est un pointeur de fonction d'arguments un pointeur de `int` et un tableau de `int`, retournant un `int`

Correction

5 `int (** x5[3])(void)(char *);`

`x5` est un tableau de 3 pointeurs de fonction sans argument retournant un pointeur de fonction d'argument un pointeur de `char`, et retournant un `int`