

المؤثرات Operators

المؤثر : هو عبارة عن رمز رياضي يقوم بإجراء عملية معينة على معاملاته والمعامل هو القيمة التي يجري عليها المؤثر العملية.

أنواع المؤثرات في لغة C++

يوجد في لغة C++ عدد من المؤثرات هي :

1. المؤثرات الحسابية Arithmetic Operators :

هناك خمسة مؤثرات حسابية في لغة C++ حيث الجدول التالي يوضح المؤثرات الحسابية الأساسية وما يكافئها في لغة C++

العملية الحسابية	في الجبر	في C++	مثال
الجمع	$a+b$	$a+b$	$c=a+b$
الطرح	$a-b$	$a-b$	$c=a-b$
القسمة	$\frac{a}{b}$	a/b	$c=a/b$
الضرب	$a \times b$	$a*b$	$c=a*b$
باقي القسمة		$a \% b$	$c=a \% b$

يجب أن نأخذ في الاعتبار في حالة القسمة (/) يكون الناتج عدداً صحيحاً إذا كانت عناصر البيانات المستعملة أعداد صحيحة وكذلك بالنسبة للمؤثر (%) يجب أن تكون عناصر البيانات قيم صحيحة وإلا فالنتيجة تكون خاطئة.

أولوية تنفيذ المؤثرات الحسابية:

أولوية التنفيذ	العملية	رمز العملية
تحسب أولاً، وإذا كانت الأقواس متداخلة فإن الأقواس الداخلية تحسب أولاً ثم التي تليها	الأقواس	()
تحسب ثانياً، إذا تعددت العمليات فإنه يتم احتسابها من اليسار إلى اليمين	الضرب والقسمة وباقى القسمة	* ، / ، %
تحسب أخيراً، إذا تعددت العمليات فإنه يتم احتسابها من اليسار إلى اليمين	الجمع والطرح	+ ، -

أمثلة:

الجدول التالي يوضح بعض العمليات الحسابية الجبرية وما يقابلها بلغة ++C

بلغة ++C	جبرياً
$m = (a + b + c + d) / 4$	$m = \frac{a + b + c + d}{4}$
$m = a + b + c + d / 4$	$m = a + b + c + \frac{d}{4}$
$m = a * b + w / x - y$	$m = ab + \frac{w}{x} - y$
$m = a + (b + c) / 8 + 5 * d - (a / b)$	$m = a + \frac{b + c}{8} + 5d - \frac{a}{b}$

مثال 1:

يوضح العلاقة بين عناصر البيانات من النوع الصحيح والمؤثرات الحسابية.

```
# include <iostream.h >
int main()
{
int a,b;
a=18;
b=5;
cout<<a+b<<"\n";
cout<<a-b<<"\n";
cout<<a*b<<"\n";
cout<<a/b<<"\n";
cout<<a%b<<"\n";
return 0;
}
```

مثال 2:

يمكن استخدام المؤثرات مع المتغيرات الصحيحة والحقيقة كما يلي :

```
# include <iostream.h >
int main()
{
int a,b;
float x,y;
a=100;
b=5;
x=9-6;
y=0-3;
cout<<x+y<<"\n";
cout<<a-b<<"\n";
cout<<a*b<<"\n";
cout<<x/y<<"\n";
return 0;
}
```

مثال 3: يوضح أولوية المؤثرات الحسابية

ما هو مخرج البرنامج التالي :

```
#include <iostream.h>
int main()
{
    int p,q,r,x;
    p=2;
    q=6;
    r=p + q / p;
    x=( p + q ) / p;
    cout<<"r = "<<r<<"\n x = "<<x<<"\n";
    return 0;
}
```

المخرج هو

$r = 5$
 $x = 4$

مثال 4:

ما هي نتيجة العمليات الحسابية التالية :

- | | |
|-----------------------------|---------|
| 1) $x=7+3*6/2-1;$ | $x=15$ |
| 2) $y=2\%2+2*2-2/2;$ | $y=3$ |
| 3) $Z=(3*9*(3+(9*3/(3))));$ | $z=324$ |

تمرين:

ما هي نتيجة العمليات الحسابية التالية:

- a) $y=2*5/2+6-12/3$
- b) $z=(5-6+8/2)+6*9/3-2$
- c) $w=10-8\%3*2+9$

2. المؤثرات العلائقية (مؤثرات المقارنة) Relational Operators :

الجدول التالي يوضح مؤثرات المقارنة وما يكافئها في لغة C++ ، والتي نستطيع استخدامها مع أي زوج من العناصر ويكون ناتجها إما صحيحاً True أو خاطئاً False .

وهي كما يلي :

مثال في لغة C++	معنى المؤثر	مؤثر المقارنة
$x == y$	يساوي	$==$
$x != y$	لا يساوي	$!=$
$x > y$	أكبر من	$>$
$x < y$	أصغر من	$<$
$x >= y$	أكبر من أو يساوي	$>=$
$x <= y$	أصغر من أو يساوي	$<=$

عند استخدام العمليات ذات الإشارات المزدوجة ($>=$, $<=$, $!=$) تجنب وضع فراغ بين علامة المساواة والعلامة الأخرى المرافقة لأن ذلك يتسبب بإظهار أخطاء قواعدية في برنامجك.



مثال:

افرض أن المتغيرين i و j من النوع الصحيح ، وحدد لهما القيمتين 5 , 3- على التوالي وفيما يلي مجموعة من التعبيرات والنتائج المناظرة لها .

القيمة	التعبير
true	$i != j$
false	$j * 2 == i * 2$
false	$i - j < i + j$
true	$i - 3 == j + 5$
true	$j < i > i + j$

3. المؤثرات المنطقية Logical Operators

وهي تستخدم لربط أكثر من تعبير منطقي بسيط مع بعضها البعض و هي :

الاسم	الاسم بالعربي	رمز العملية
and	و	& &
or	أو	
not	النفى	!

وأولوية تنفيذها من اليسار إلى اليمين كما يلي:

NOT → AND → OR

والجدول التالي يوضح جدول الصواب والخطأ لهذه المؤثرات:

X	Y	X && Y	X Y	!X
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

مثال: ما هو ناتج التعبيرات التالية :

1- ((5==5)&&(3>6))

True && False
False

2- ((5==5)|| (3>6))

True || False

True

F && F || F || F && F-3

F && F || F || F && F

F || F || F

F || F

F

4. المؤثر الشرطي (?:) Conditional Operator

يستخدم لاختبار شرط معين ويرجع قيمة إذا تحقق الشرط ويرجع قيمة أخرى إذا لم يتحقق الشرط .
والصيغة العامة له هي :

(condition) ? result1 : result2

إذن المؤثر الشرطي يقبل ثلاث معاملات هي

- الشرط المنطقي (condition)
- القيمة الراجعة إذا تحقق الشرط result1
- القيمة الراجعة إذا لم يتحقق الشرط result2

مثال 1:- برنامج لإيجاد القيمة الأكبر من بين قيمتين باستخدام المؤثر الشرطي

```
#include <iostream.h>
int main ( )
{
    int a,b,c;
    a=2;
    b=7;
    c=(a>b)?a:b;
    cout<<"max = "<<c;
    return o;
}
```

مثال 2:- عدل البرنامج السابق لإدخال ثلاثة قيم وإيجاد القيمة الأكبر

```
#include <iostream.h>
int main ( )
{
    int a,b,c,d,e;
    cin>>a>>b>>c;
```

```
d=(a>b)?a:b;
e=(c>d)?c:d;
cout<<"max = "<<e;
return o;
}
```

5. مؤثرات التعيين المركبة compound assignation operators

(+=, -=, *=, /=, %=)

يتوفر في لغة ++C عدداً من عمليات الإسناد. فبالإضافة لعملية الإسناد العادية التي تستخدم فيها الإشارة = يوجد طرق مختصرة أخرى كما في الأمثلة التالية:

- عملية الإسناد $b=b+c$ تكافئ عملية الإسناد المختصرة $b+=c$
- عملية الإسناد $b=b-c$ تكافئ عملية الإسناد المختصرة $b-=c$
- عملية الإسناد $b=b*c$ تكافئ عملية الإسناد المختصرة $b*=c$
- عملية الإسناد $b=b/c$ تكافئ عملية الإسناد المختصرة $b/=c$
- عملية الإسناد $b=b\%c$ تكافئ عملية الإسناد المختصرة $b\%=c$

وبطريقة أخرى يمكن كتابة عمليات الإسناد والتخصيص كما:

$v+=r$ تجمع r إلى v وتضع النتيجة في v

$v-=r$ تطرح r من v وتضع النتيجة في v

$v*=r$ تضرب r بـ v وتضع النتيجة في v

$v/=r$ تقسم v على r وتضع النتيجة في v

$v\%=r$ تجد باقي قسمة v على r وتضع النتيجة في v

مثال 1:-

التعبير
value+=increase
a-=5
a/=b
price*=units+1

التعبير المكافئ
Value=value + increase
a=a-5
a=a/b
price=price*(units + 1)

مثال 2:-

```
//compound assignation
#include <iostream.h>
int main ( )
{
int a,b=3;
a=b;
a+=2;      //equivalent to a=a+2
cout<<a;
return 0;
}
```

مثال 3:- ما هو ناتج البرنامج التالي :

```
#include <iostream.h>
int main ( )
{
int p,q,r;
p=2;
q=3;
r=10;
r/=p+q;
cout<<"r="<<r;
return 0;
}
```

6. مؤثرات التزايد والتناقص Increase and Decrease Operators

مؤثر التزايد هو ++ ويعني زيادة قيمة المتغير بمقدار واحد
ومؤثر التناقص هو - ويعني إنقاص قيمة المتغير بمقدار واحد

مثال 1:- العبارة C++

تعني زيادة قيمة المتغير C بمقدار واحد وهي مكافئة للتعبير التالية

C+=1
C=C+1

وينطبق هذا أيضا في حالة النقصان (--)

والتزايد والتناقص يمكن أن يكون قبلي أو بعدي, أي يمكن وضع ++ أو -- في بداية المتغير أو في نهايته
ولكن هناك فرق كبير بينهما .

فالتعبير $i++$ يعني إضافة واحد لقيمة المتغير i قبل احتساب التعبير
 في حين أن التعبير $++i$ يعني استخدام القيمة الحالية للمتغير i ثم إضافة واحد إلى المتغير
 أي أن عند استعمال $++i$ نزيد قيمة i قبل احتساب التخصيص وعند استعمال $i++$ يحسب التخصيص قبل زيادة i
 وينطبق هذا أيضا في حالة النقصان $--$

وباختصار يمكن تمثيل توضيح التزايد والتناقص القبلي والبعدي كما يلي

- $++x$ تزايد واحداً إلى x وبعدها تستعمل العبارة محتوية x الجديد.
- $x++$ يستعمل محتوية x أولاً وبعدها يزداد إليه واحد.
- $--y$ تنقص واحداً من y وبعدها تستعمل العبارة محتوية y الجديد.
- $y--$ يستعمل أولاً محتوية y وبعدها ينقص منه واحد.

مثال 2:

ما هي نتيجة كل من a ، b بعد كل عملية من العمليات الحسابية التالية :

$$a = 6 \quad b = 6$$

1. في البداية

$a = + + b$	$a=7 \quad b=7$
$a = b ++$	$a=7 \quad b=8$
$++ b$	$a=7 \quad b=9$
$a = - - b$	$a=8 \quad b=8$
$a = b - -$	$a=8 \quad b=7$

2. إذا كان في البدء

$$a = 2, \quad b = 4, \quad c = 6$$

a, b, c بعد كل من العبارات التالية :

ما القيمة التي سيأخذها

1. $a += 5 + a$
2. $a * = b + + - c$
3. $a - = -b + + * ++ c$

الحل :

(1)

$$a += 5 + a = 5 + 2 = 7$$

$$a = a + 7 = 2 + 7 = 9$$

a=9,b=4, c=6

القيم الجديدة

(2)

$$a * = b + + - c = 4 - 6 = -2$$

$$a = a * -2 = 9 * -2 = -18$$

a = -18 , b = 5 , c = 6

القيم الجديدة

(3)

$$a - = -b + + * + + c$$

$$a - = -5 * 7 = -35$$

$$a = a - (-35) = -18 + 35 = 17$$

a = 17, b = 6, c = 7

القيم الجديدة

مثال:- ما هو ناتج البرنامج التالي :

```
#include <iostream.h>
int main ( )
{
    int i,j,k;
    i=j=2;
    k=++i;
    cout<<"i="<<i<<"k="<<k<<"\n";
    k=j++;
    cout<<"j="<<j<<"k="<<k<<"\n";
    return 0;
}
```

الناتج هو :-

i=3 k=3

j=3 k=2