المحاضرة الثالثة
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	aXb	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:

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

أمثلة:

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

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

مثال:

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:

يمكن استخدام المؤثرات مع المتغيرات الصحيحة والحقيقة كما يلي :

```
# include <iostream.h >
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: 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 1:

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:

- y=2*5/2+6-12/3
- z=(5-6+8/2)+6*9/3-2
- 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 :

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

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

مثال:

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

التعبير	القيمة
I != J	TRUE
J * 2 = = I * 2	FALSE
I – J < I + J	FALSE
I – 3 = = J + 5	TRUE
J < I > I + J	TRUE

3- المؤثرات المنطقية logical operators

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

رمز العملية	الاسم بالعربي	الاسم
& &	و	and
||	أو	or
!	النفي	not

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 التعابير التالية :

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

True && False

False

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

True || False

True

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

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

مثال1:- 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;
 }
```

مثال2:- عدل البرنامج السابق لإدخال ثلاثة قيم و إيجاد 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 vعلى r وتضع النتيجة في v

v٪= r تجد باقي قسمة v على r وتضع النتيجة في v

مثال1:-

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

مثال2:-

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

مثال3:- ما هو ناتج البرنامج التالي :

```
#include <iostream.h>
main ()
```

```
{
int  p,q,r;
p=2;
q=3;
r=10;
r/=p+q;
cout<<"r="<<r;
return 0;
```

6- مؤثرات التزايد و التناقص Increase and decrease operators

مؤثر التزايد هو ++ ويعني زيادة قيمة المتغير بمقدار واحد

ومؤثر التناقص هو – ويعني إنقاص قيمة المتغير بمقدار واحد

مثال1 :- العبارة C++

تعني زيادة قيمة المتغير C بمقدار واحد وهي مكافئة للتعابير 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 .

فالتعبيرi ++ 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 فبل احتساب التخصيص وعند استعمال ++i يحسب التخصيص قبل زيادةi

وينطبق هذا أيضا في حالة النقصان - -

وباختصار يمكن يمكن توضيح التزايد والتناقص القبلي والبعدي كما يلي

- x++ تزيد واحداً إلى x وبعدئذ تستعمل العبارة محتوى x الجديد.
- ++x يستعمل محتوى x أولاً وبعدئذ يزاد إليه واحد.
- y-- تنقص واحداً من y وبعدئذ تستعمل العبارة محتوى y الجديد.
- --y يستعمل أولاً محتوى y وبعدئذ ينقص منه واحد.

مثال2 :

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

1. في البداية a = 6 b = 6

a =+ +b	a=7 b=7
a =b ++	a=7 b=8
++ b	a=7 b=9
a = - -b	a=8 b=8
a = b- -	a=8 b=7

2. إذا كان في البدء

a = 2, b = 4, c = 6

ما القيمة التي سيأخذها a, b,c بعد كل من **العبارات** التالية :

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

القيم الجديدة a=9,b=4, c=6

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

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

القيم الجديدة a =-18 , b = 5 , c = 6

3)

a- = -b+ +*++c

a- = -5*7=-35

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

القيم الجديدة a =17, b =6, c=7

مثال:- **ما هو ناتج البرنامج التالي :**

```
#include <iostream.h>
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