

## فصل دوم

ساختارهای تصمیم گیری و تکرار

I . ASHKAVAND RAD



## فهرست مطالب فصل دوم

۱. عملگر های رابطه ای
۲. عملگر شرطی
۳. دستورالعمل شرطی
۴. عملگر کاما
۵. عملگر های منطقی
۶. دستورالعمل For



## عملگرهای رابطه ای

از این عملگرها برای تعیین اینکه آیا دو عدد با هم معادلند یا یکی از دیگری بزرگتر یا کوچکتر می باشد استفاده می گردد. عملگرهای رابطه ای عبارتند از:

==	مساوی
!=	مخالف
>	بزرگتر
>=	بزرگتر یا مساوی
<	کوچکتر
<=	کوچکتر یا مساوی



# عملگر شرطی

شکل کلی عملگر شرطی بصورت زیر می باشد:

```
expression _ test ? expression _ true : expression _ false
```

عملگر شرطی تنها عملگری در **C++** می باشد که دارای سه عملوند می باشد.



مثال ۱ :

```
int x=10,y=20,b;  
b=(x>y) ? x : y ;
```

این دو دستور العمل باعث میشوند که ماکزیمم مقادیر  $x$  و  $y$  در  $b$  قرار بگیرد.

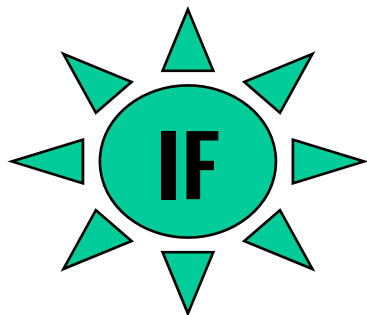
---

مثال ۲ :

```
x>=10 ? cout << "passed" : cout << "failed" ;
```

اگر مقدار  $x$  بزرگتر یا مساوی ده باشد رشته `passed` در غیر اینصورت رشته `failed` نمایش داده میشود.





## دستور العمل شرطی

توسط این دستور شرطی را تست نموده و بسته به آنکه شرط درست یا غلط باشد عکس العمل خاصی را نشان دهیم.

```
if( عبارت )  
{  
    دستورالعمل 1  
    .  
    دستورالعمل n  
}  
else  
{  
    دستورالعمل 1  
    .  
    دستورالعمل n  
}
```



مثال ۱ :

```
if(x != y)
{
cout << x ;
++ x ;
}
else
    {
cout << y ;
-- y ;
    }
```



I . ASHKAVAND RAD

## مثال ۲:

برنامه زیر یک عدد اعشاری را از ورودی گرفته جذر آنرا محاسبه می نماید.

```
#include <iostream.h>
#include <math . h>
int main( )
{
float x,s;
cin >> x ;
if( x < 0 )
cout << " x is negative" << endl ;
else
{
s = sqrt(x) ;
cout << s << endl ;
}

return 0;
```



## عملگر کاما

تعدادی عبارت را می توان با کاما بهم متصل نمود و تشکیل یک عبارت پیچیده تری را داد. این عبارتها به ترتیب از چپ به راست ارزیابی شده و مقدار عبارت معادل عبارت  $n$  می باشد.



(عبارت  $n$ , ..., عبارت 3, عبارت 2, عبارت 1)

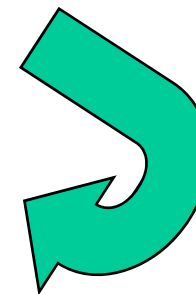


مثال :

اگر داشته باشیم `int a=2 , b=4 , c=5 ;` عبارت زیر را در نظر بگیرید:

`(++ a , a+b, ++ c, c+b)`

مقدار عبارت برابر است با `b+c` که معادل 10 می باشد.



# عملگرهای منطقی

با استفاده از عملگرهای منطقی می توان شرطهای ترکیبی در برنامه ایجاد نمود.  
عملگرهای منطقی عبارتست از :

AND

OR

NOT

که در C++ به ترتیب بصورت زیر نشان داده میشود.

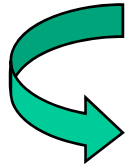
&&

||  
!



## جدول درستی سه عملگر شرطی

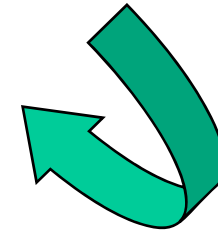
# And



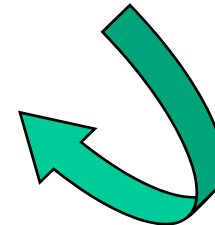
a	b	a && b
true	true	True
true	false	False
false	true	False
false	false	False

a	b	a    b
true	true	True
true	false	True
false	true	True
false	false	False

# Or



# Not

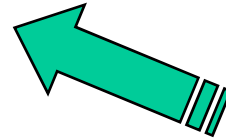


a	!a
true	False
false	True



## چند مثال :

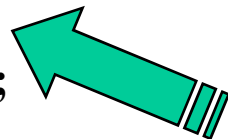
```
if ((x == 5) || (y != 0))  
    cout << x << endl ;
```



اگر  $x$  برابر با 5 یا  $y$  مخالف صفر باشد مقدار  $x$  نمایش داده شود .

---

```
if(x)  
    x = 0 ;
```



اگر مقدار  $x$  مخالف صفر باشد، آنگاه  $x$  برابر با صفر شود .



برنامه زیر طول سه پاره خط را از ورودی گرفته مشخص می نماید که آیا تشکیل یک مثلث میدهد یا خیر؟

```
#include <iostream.h >
int main( )
{
float a, b, c;
cout << "Enter three real numbers" << endl ;
cin >> a >> b >> c; //
if(( a < b + c) &&(b < a+c) &&(c < a+b))
cout << "It is a triangle" ;
else
cout << "Not a triangle" ;
return 0 ;
}
```



I . ASHKAVAND RAD

# دستور العمل For

از دستور العمل **for** برای تکرار دستورالعملها استفاده میشود. شکل کلی دستور **for** بصورت زیر می باشد:

```
(عبارت 3 ; عبارت 2 ; عبارت 1) for
{
  دستورالعمل 1 ;
  دستورالعمل 2 ;
  .
  .
  .
  دستورالعمل n ;
}
```



برنامه زیر عدد صحیح و مثبت  $n$  را از ورودی گرفته فاکتوریل آنرا محاسبه و نمایش می دهد.

```
#include <iostream.h>
int main( )
{
int n, i ;
long fact = 1 ;
cout << "Enter a positive integer number";
cin >> n;
for( i=1; i<=n; ++i) fact *= i;
cout << fact << endl;
return 0 ;
}
```



برنامه زیر مجموع اعداد صحیح و متوالی بین ۱ تا n را محاسبه نموده و نمایش می دهد.

```
#include <iostream.h>
int main( )
{
int n, i=1 ;
long s = 0 ;
cin >> n ;
for(; i<=n; i++) s += i;
cout << s ;
return 0 ; }
```



برنامه زیر ارقام 0 تا 9 را نمایش می دهد.

```
#include <iostream.h>
int main( )
{
int j=0 ;
for( ; j <= 9 ; ) cout << j++ << endl;
return 0 ;
}
```



I . ASHKAVAND RAD

برنامه زیر کلیه اعداد سه رقمی که با ارقام 1 ، 2 ، 3 ایجاد می شوند را نمایش می دهد.

```
#include <iostream.h>
int main( )
{
int i,j,k,n;
for(i=1; i<=3; ++i)
for(j=1; j<=3; ++j)
for(k=1; k<=3; ++k)
{
n=i*100 + j*10+k;
cout << n << '\n' ;
}
return 0 ;
}
```



I . ASHKAVAND RAD