

نام خدا



سری های آموزشی آشنایی با زبان برنامه نویسی C++

قسمت دوم : مبانی C++

عملگر های ریاضی در C++ :

عملگر های ریاضی در C++ به صورت زیر تعریف می شوند :

نام عملگر	علامت عملگر در زبان C++
جمع	+
تفریق	-
ضرب	*
تقسیم	/
باقیمانده	%

بدیهی است که حاصل این عملگر ها می تواند صحیح یا اعشاری باشد .

در زبان C++ برای انجام عملیات جمع و تفریق با 1 دو عملگر دیگر نیز وجود دارد که هدف از ایجاد آنها عموم راحت تر کردن کار بوده است.

نام عملگر	علامت عملگر در زبان C++
جمع خود محتوای متغیر با 1	++
کم کردن 1 واحد از محتوای متغیر	--

نکته ی مهم در استفاده از این دو عملگر اخیر توجه به نقش های مختلف آن در اولویت های مختلف است . یعنی اینکه ++a با a++ تفاوت خواهد داشت . و عملگر -- نیز به همین ترتیب است .

برای اینکه بهتر متوجه منظورم بشوید به مثال های زیر توجه فرمایید :

```
#include <iostream.h>

int main()
{
    int c;

    c = 5;
    cout << c << endl;      // print 5
    cout << c++ << endl;    // print 5 then postincrement
    cout << c << endl << endl; // print 6

    c = 5;
    cout << c << endl;      // print 5
    cout << ++c << endl;    // pre increment then print 6
    cout << c << endl;      // print 6

    return 0;
}
```

در خروجی برنامه ی فوق خواهیم داشت :

```
5
5
6

5
6
6
```

توجه : endl خط فرمان را سطر پایین تر می آورد . فرق آن با \n در این است که \n در داخل " " نوشته می شود اما endl خودش به تنهایی نگاشته می شود .

مسئله ی 1 : برنامه ای بنویسید که دو عدد را گرفته و مجموع آنها را در خروجی چاپ کند .

```
// program for suming up twe numbers !
#include <iostream.h>
int main()
{
int num1,num2,sum;
cout<<"please enter the first integer number ! : " ;
cin>>num1;
cout<<"please enter the second integer number ! : " ;
cin>>num2;
sum=num1+num2;
cout<<" The sum is "<<sum<<" \n";
return 0;
}
```

در بحث عملگر های ریاضی اولویت بعدی عملگر ها خود نیز مسئله ی مهمی به شمار می آید. اولویت عملگر ها به صورت زیر است :

ترتیب ی اولویت	عملگر ها
1	() پرانتز ها
2	/ یا *
3	%
4	- یا +

چند نمونه دیگر از عملگر های ریاضی که برای آسانی کار ارائه شده اند :

علامت عملگر	مثال	شکل دیگر عملگر
+=	a+=5	a=a+5
-=	a-=5	a=a-5
=	a=5	a=a*5
/=	a/=5	a=a/5
%=	a%=5	a=a%5

برای اینکه با طرز کار با این عملگرها آشنا شوید ، به مثال زیر توجه فرمایید :

```
#include <iostream.h>

int main()
{
int a;

a=6;
a+=6;
cout<<a<<endl;
a=6;
a-=6;
cout<<a<<endl;
a=6;
a*=6;
cout<<a<<endl;
a=6;
a/=6;
cout<<a<<endl;
a=6;
a%=6;
cout<<a<<endl;

return 0;
}
```

مسئله ی 2 : برنامه ای بنویسید که حقوق پایه و تعداد فرزندان یک کارگر را از ورودی گرفته و حقوق کل وی را از فرمول زیر بدست آورد :

$$10 * \text{فرزندان} + \text{حقوق پایه} = \text{حقوق کل}$$

```
#include <iostream.h>

int main()
{
    int child,salary,wholesalary;

    cout<<" Enter yor child's number: ";

    cin>>child;

    cout<<" Enter your salary :";

    cin>>salary;

    wholesalary=salary+10*child;

    cout<<"your whole salary is : "<<wholesalary<< endl;

return 0;
}
```

آرایه های چند وجهی :

آرایه های چند بعدی ، نوع پیشرفته تری از آرایه ها هستند که می توانند اطلاعات بیشتری را در خود ذخیره کنند . و در عوض نیز کار های پیشرفته تری را انجام دهند . (فعلا تا همین کافی است ! در قسمت های بعد بیشتر درباره کاربرد های اینها آشنا خواهید شد .)

ساختار کلی برای استفاده از آرایه های چند بعدی به صورت زیر است :

```
Type Arrayname
[size1][size2]...[sizeN];
```

برای مثال در در نمونه ی زیر آرایه ی kami دارای 2 بعد می باشد که دارای 2 سطر و ستون است . می خواهیم مقادیر اولیه ی این آرایه را بگیریم و 1 مقدار به آنها اضافه کنیم و بعد مقادیر آرایه ی kami را چاپ کنیم :

```
#include <iostream.h>

int main()
{

int kami[2][2];

cin>>kami[1][1];
cin>>kami[2][1];
cin>>kami[1][2];
cin>>kami[2][2];

kami[1][1]++;
```

```
kami[2][1]++;
kami[1][2]++;
kami[2][2]++;

cout<<"kami [1][1]:
"<<kami[1][1]<<endl;
cout<<"kami [2][1]:
"<<kami[2][1]<<endl;
cout<<"kami [1][2]:
"<<kami[1][2]<<endl;
cout<<"kami [2][2]:
"<<kami[2][2]<<endl;

return 0;

}
```

عملگر های منطقی :

در جدول زیر انواع عملگر های مقایسه ای یا منطقی را مشاهده می فرمایید:

نام عملگر	علامت عملگر در زبان C++
AND	&&
OR	
NOT	!
کوچکتر	<
کوچکتر یا مساوی	<=
بزرگتر	>
بزرگتر یا مساوی	>=
مقایسه	==
نا مساوی	!=
انتصاب شرطی	?:

بیشترین استفاده ی از عملگر های منطقی یا مقایسه ای در ساختار های شرطی است . برای آشنایی با ساختار های شرطی به مطلب بعدی توجه فرمایید .

ساختار های تصمیم گیری در C++ :

- 1- دستور if-else: زمانی از این ساختار استفاده می شود که شرط ها کم باشد.
- 2- دستور switch-case: زمانی از این ساختار استفاده می شود که تعداد تصمیم گیری ها زیاد باشد .

ساختار شرطی if :

ساختار کلی شرطی if به صورت زیر است :

```
If (condition1)
{
    دستورات قسمت اول ;
}
Else
{
    دستورات قسمت دوم ;
}
```

توجه کنید که ساختار بالا یک ساختار کلی می باشد و ممکن است در حالات شکل آن تغییر کند . مثل حالات زیر ، آنها را به خاطر بسپارید ! :

1- دستورات قسمت اول یا دستورات قسمت دوم یا هر دو فقط شامل ی دستور باشند :

```
If (condition1)
    دستور قسمت اول ;
Else
{
    دستورات قسمت دوم ;
}
```

```
If (condition1)
{
    دستورات قسمت اول ;
}
Else
    دستور قسمت دوم ;
```

```
If (condition1)
    دستور قسمت اول ;
Else
    دستور قسمت دوم ;
```

2- در بعضی مواقع استفاده از قسمت دوم این ساختار تصمیم گیری (else) لازم نیست . یعنی اینکه شما فقط قصد استفاده از قسمت if را دارید :

```
If (condition)
{
    دستورات ;
}
```

توجه کنید که در صورتی هم که دستورات شما شامل فقط یک دستور باشد ، لازم نیست که از دو آکولاد استفاده کنید :

```
If (condition)
دستور ;
```

مسئله 3 : برنامه ای بنویسید که 2 عدد دریافت کند و بزرگترین آنها را بنویسد .
جواب :

```
#include <iostream.h>
int main()
{
    int num1,num2;
    cout<<" Enter your first number: ";
    cin>>num1;
    cout<<" Enter your second number : ";
    cin>>num2;

    if (num1>num2)
    cout<<num1<<" is bigger ! ";
    else
    cout<<num2<<" is bigger ! ";

return 0;
}
```

مسئله 4 : برنامه ای بنویسید که یک عدد را از ورودی گرفته و مشخص کند که آن عدد زوج است یا فرد .

جواب :

```
#include <iostream.h>
int main()
{
    int num;
    int rest;

    cout<<" Enter your number: ";
    cin>>num;
    rest=num % 2;

    if (rest!=0)
    cout<<num<<" is fard(odd) ! ";
    else
    cout<<num<<" is zoj(even) ! ";

return 0;
}
```