جلسه ی سوم(تصمیم گیری وتکرار)

عملگر های رابطه ای

این عملگر ها دو مقدار را به هم مقایسه نموده و حاصل را به صورت true یا false برمی گردانند

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

عملگر شرطی

دستورات...درست بودن شرط ; دستورات اجرایی در صورت درست بودن شرط ?عبارت مورد ارزیابی
(3>1)?0:1;

برای مثال حاصل عبارت بالا صفر است
میتوان بجای صفر ویک متغیر یا عبارت محاسباتی قرارداد
یا دستور دیگری مثل چاپ یک عبارت قرار داد که در این صورت این عملگر
دیگر مقداری را بر نمی گرداند.البته بازم می تونه مقداربر گردونه ....

عملگر کاما

این عملگر ربطی به این مبحث نداره اینجا میگم چون خیلی جا ها می شه استفادش
کرد واگه بلدش نباشید حسابی گیج می شید.منم حالش رو ندارم ده بار جاهای مختلف
توضیح بدم .خیلی سادس اما توی دستورات مختلف می تونه گیج کننده باشه.
(عبارت1,عبارت2,...,عبارت n);
این دستور تمام عبارات را اجرا نموده وارزش عبارت یعنی مقدار بازگشتی آن
برابر با سمت راست ترین عبارت است.
(a++,b=a,a);
خب حالا اگر در دستور عملگر شرطی از کاما استفاده کنیم می شود هم دستور اجرا کرد هم مقدار بر گرداند

بلاک های دستور

هر بلاک دستور قسمتی از برنامه است که کامپایلر آنها را با هم فرض می کند
دو نوع بلاک داریم نوع اول بلاک تک دستوری است که فقط یک دستور است
وبه یک سمیکالن ختم می شود نوع دوم بلاک با بیش از یک دستور است که با { شروع وبه} ختم می شود وتمام دستور های مد نظر بین این دو قرار می گیرند

دستور if

این دستور ارزش یک عبارت را ارزیابی نموده بر اساس آن دستوراتی را اجرا می نماید

```
If(عبارت)
دستور در صورت صحیح بودن ارزش عبارت ;
else
دستور در صورت نادرست بودن ارزش عبارت ;
```

این حالت بلاک ها تک دستوری میباشد هر کدام از این بلاک ها می توانند چند دستوری باشند

```
If(عبارت){
```

cpplearning.blogfa.com

```
.
.
.
}
else
{
.
.
.
}
```

همچنین بلاک else می تواند وجود نداشته باشد عبارت مورد ارزیابی هم میتواند یک عدد یک مقایسه ساده یا یک مقایسه ی مرکب از عملگر های رابطه ای و منطقی باشد

نکته

ارزش اعداد بزرگتراز صفر صحیح است

نکته

ارزش عبارت b=a برابر به مقدار متغیر a است

مثال چاپ قدر مطلق

```
if(a>0)
cout<<a;
else
cout<<-a;
```

مثال چاپ و محاسبه ی قدر مطلق

```
If(a>0)
cout<<a;
else{
cout<<-a;
a=-a;}
```

حلقه ی for

حلقه ای با یک یا چند اندیس.

```
for(مقدار اولیه:شرط حلقه: گام حلقه)
بلاک دستور تک دستوری یا چند دستوری
```

شرط حلقه تا زمانی که درست هست حلقه اجرا میشه شرط حلقه را میتوان به هر صورت تعریف کر حتی می شود شرطی استفاده کرد که هیچ ربطی به اندیس حلقه نداشته باشه در حالی که پاسکال شرط حلقه را خودش در نظر میگیرد و شما حتی نمی توانید شرط را ببینید مبتد ی ها هم که متوجه وجود چیزی به نام شرط نمی شوند بعد از شرط حلقه می رسیم به گام حلقه شما می توانی با دستورات محاسباتی تعریف کنید که اندیس حلقه چه مقدار تغییر کند حال آنکه در پاسکال مقدار پیش فرض یک داره!!! هیچ کسی هم نمیبیندش تا زه گام حلقه و مقدار اولیه می توانند با استفاده از عملگر کاما چند تا یی باشند مثلا می شه دوتا متغییر رو مقدار اولیه داد بعد دو تا گام حلقه تعریف کرد و در نتیجه حلقه دو اندیسی تولید کرد!!! کف کردید تازه میشه مقدار اولیه نداد شاید برنامه جوری بود که

cpplearning.blogfa.com

حلقه باید از مقدار فعلی متغییر اندیس حلقه شروع به کار کند.حالا مثال ها رو ببینید و بازم کف کنید.

```
for(i=0;i<10;i++)
cout<<i<<endl;

for(int i=0;i>-10;i--)
cout<<i<<endl;

for(i=0,j=32;j>=0;j/=2,i+=j){
cout<<i<<endl;
cout<<j<<endl;
}

for(;i>-10;i--)
cout<<i<<endl;

for(int i=0;i>-10;){
cout<<i<<endl;
i--;
}
```

حتی می شود حلقه while را باforشبیه سازی کرد

حلقه ی while

شرط این حلقه در ابتدای هر بار رجوع به ابتدای حلقه بررسی می شود
هر گاه شرط نا درست شود اجرا حلقه خاتمه می یابد

```
while(شرط حلقه)
بلاک دستور تک دستوری یا چند دستوری

while(a>0){
a--;
cout<<a<<endl;
}
```

حلقه بینهایت به همراه بوق متد و مانیتوری پر از پیغام خطا برای خروج از کنترل پاوز استفاده کنید.

```
while(1)
cout<<"Error!"<<'\a';
```

حلقه ی do while

شرط این حلقه در ابتدای هر بار رجوع به انتهای حلقه بررسی می شود
محتویات بلاک دستور این حلقه حداقل یک با اجرا می شود.
هر گاه شرط نا درست شود اجرا حلقه خاتمه می یابد.

```
do
بلاک دستور تک دستوری یا چند دستوری
while(شرط حلقه);
```

دستورbreak

این دستور برای پرش به بیرون از حلقه ها استفاده می شود بدلایلی بهتر است استفاده نشود و بجایش شرط حلقه را نادرست کرد اساسا دستورات پرش دار باعث کاهش سرعت اجرا می شوند.

cpplearning.blogfa.com

دستور continue

این دستور باعث پرش به ابتدای حلقه می شود بجای این دستور هم بهتر است بلاک های برنامه را بگونه ای نوشت که نیاز به استفاده این دستور نباشد.

دودستور اخیر اگر درون حلقه های تودر تو استفاده شوند روی همان حلقه ای عمل می کنند که درون بلاک دستور آن حلقه استفاده شده اند.

دستور switch

گاهی اوقات لازم است برای مقادیر مختلف یک عبارت تصمیمات مختلفی گرفته شود در این موارد استفاده از ifمی تواند سر در گم کننده باشد.

```
switch(عبارت){
        case  1مقدار:
یک یا چند دستور
        break;
        case  2مقدار:
یک یا چند دستور
        break;
          .
          .
          .
        default:
یک یا چند دستور
}
```

این ساختار می تواند فاقد defaultباشد دستورات این بخش در صورت درست نبودن هیچ یک از مقادیر قبلی اجرا می شود

مقادیر موجود در caseها نمی توانند برابر باشند(دو یا چند caseبا یک مقدار)

این دستور فقط تساوی عبارت با مقادیررا مور بررسی قرار میدهد.
اگر در یک caseدستور breakاستفاده نشود با caseبعدی orمی شود.
در هر case می توان ساختار switchدیگری ایجاد کرد

عملگرهای منطقی

می توانید با عمگرهای منطقی عبارات شرطی پیچیده تری ایجاد کنید
این عملگرها شامل and or not هستند که در C++ معادلشان
! || && میباشد
به مثال توجه کنید

```
if((3>a)&&(a>0))
```

اگر aبین 3و0 باشد عبارت صحیح می شود و دستورات درون بلاک ifاجرا میشود

cpplearning.blogfa.com