

جلسه ی دوم (ساختار برنامه)

زبان ++C یک زبان ساختیافته شئی گراست (ساختیافته یعنی از توابع یا قطعه برنامه های مختلف در کنار یکدیگر استفاده می کند و شئی گرا یعنی از توابع بسته بندی شده استفاده میکند بعدا توضیحات بیشتری ارائه می شود) این زبان تعدادی توابع و اشیا پیش ساخته دارد که درون فایل های سرآیند تعریف شده اند برای استفاده از این امکانات پیش ساخته باید نام این فایل ها با استفاده از دستور #include به برنامه اضافه نمود

<نام فایل>#include

فایل های سرآیند فایل‌هایی هستند با پسوند h شما میتوانید این فایل‌ها را در پوشه ی کامپایلر پوشه ی include بیابید.
ساختار کلی برنامه در ++C

<فایل سرآیند>#include

```
int main()
```

```
{
```

```
    اعلان متغیرهای محلی
```

```
    دستورات اجرایی
```

```
    return 0;
```

```
}
```

در این ساختار تابع main تعریف شده. نقطه ی شروع اجرا برنامه همین تابع است برای شروع اجرای برنامه سیستم عامل این تابع را اجرا میکند سیستم عامل پس از خاتمه ی اجرا برنامه می خواهد بداند که برنامه درست اجرا شده یانه خاتمه ی برنامه دقیقا وقتی است که اجرا تابع main خاتمه می یابد میدانیم که هر تابع می توان مقدار را به برنامه اجرا کنند باز گرداند پس تابع main می تواند مقداری را به سیستم عامل برگرداند در ساختار بالا نوع مقدار باز گشتی تابع مذکور صحیح اعلام شده آخرین دستور آن نیز دستور return است که اجرای تابع را خاتمه داده و مقدار صفر را به برنامه ی اجرا کننده (در اینجا سیستم عامل) باز می گرداند باین کار سیستم عامل از اجرا درست برنامه اطمینان می یابد.

توجه داشته باشید که دستوراتی که با # شروع می شوند دستورات پیش پردازشی هستند و به سمیکال ختم نمی شوند این دستورات قبل از ترجمه برنامه به زبان ماشین تغییراتی در برنامه ایجاد میکنند.

ورودی و خروجی

اشیا cin و cout از اشیا پیش ساخته می باشند که توابع مختلفی برای مقاصد ورودی خروجی در اختیار برنامه نویس قرار می دهند اشیا مذکور در فایل سرآیند iostream.h تعریف شده اند بنابراین برای استفاده از آنها باید این فایل را با استفاده از دستور #include به برنامه اضافه نمود.

چند مثال

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

```
int main(){
```

```
    int a;
```

```
    cout<<a;
```

```
    cin>>a;
```

```
    cout<<a*a;
```

```
    return 0;
```

```
}
```

قبلا گفتم که ++C به متغیرها مقدار اولیه نمی دهد اگه این برنامه رو اجرا کنید اول محتویات متغیره چاپ میشه یعنی همان چیزی که تو حافظه به صورت تصادفی وجود داشته اما اگه همین کار رو با پاسکال انجام بدیت نباید انتظاری غیر از چاپ شدن صفر داشت پس همیشه باید دقت داشته باشید که یه وقتی متغیری بدون مقدار اولیه توی شرطها قرار نگیره چرا که نتیجه نامعلوم یا در اصطلاح نا معتبر است

بعد از چاپ محتویات متغیره با دستور cin مقدار متغیر مذکور از کیبرد خوانده شده و دستور بعد تو آن را محاسبه نموده و چاپ میکند

کاراکترهای خاص

کاراکترهای خاصی به نام کاراکترهای کنترلی وجود دارند که یا کاراکتر خاصی را چاپ می کنند یا عمل خاصی انجام میدهند این کاراکترها با \ شروع می شوند اگر چه با بیش از یک کاراکتر نوشته می شوند اما یک کاراکتر محسوب می شوند بنابراین برای چاپ یا انتساب این کاراکترها به متغیرها باید آنها را بین دو کوتیشن تکی ('') قرار داد. البته از کوتیشن دوتایی هم میتوان استفاده نمود با این تفاوت که که عبارت حاصل یک رشته ی یک کاراکتری است و نه یک کاراکتر

عمل	کاراکتر
انتقال کنترل به صفحه جدید	\f
انتقال کنترل به خط جدید	\n
انتقال به 8 کاراکتر بعد از موقعیت جاری	\t
به صدا در آمدن بوق سیستم	\a
Backspace	\b
چاپ کوتیشن دوتایی	\"
چاپ کوتیشن تک	'
چاپ بک اسلش	\\
NULL	\0
انتقال کنترل به 8 سطر بعد	\v
چاپ دو نقطه	\:
چاپ علامت سوال	\?

مثال

```
cin>>a>>b;
cout<<a<<'\n';
cout<<b<<'\n';
cout<<'\a';
```

دراین مثال ابتدا مقادیر با cin خوانده می شوند و بعد در خطوط جدا چاپ می شوند در پایان سیستم بوق می زند بجای '\n' می توان از endl استفاده نمود

```
cout<<a<<endl;
```

در ++C رشته ها را بین کوتیشن دوتایی قرار می دهند پس برای چاپ رشته باید بنویسیم

```
cout<<"C++ is a powerful language";
```

C++ & سالاد ونوشابه

نویت رسیده به چند تا دستور با حال تا راحت تر بتونیم برنامه بنویسیم اولیش تابع getch() برنامه وقتی به این تابع می رسه منتظر می مونه تا کاربر یک کلید رو فشار بده تازه کد کلیدی که فشرده شده رو هم بر میگردونه می تونید از این تابع در انتهای برنامه استفاده کنید تا بتونید خروجی برنامه رو قبل از

خروج از صفحه ببینید.
دومیش تابع clrscr() که خلیم تابلو که صفحه نمایش رو پاک میکنه
سومیش تابع gotoxy(int x,int y) که موقعیت جاری مکان نامرو تغییر می ده
تمامی این توابع در فایل سرآیند conio.h قرار دارند. می دونید که معنیش چیه؟
اگه نمیدونید!!!! این جلسه رو از اول بخونید.

```
#include<iostream.h>
#include<conio.h>

int main(){
char kb;
cout<<"ye mesal";
gotoxy(10,5);
cout<<"*****";
kb=getch();
cout<<"You Pressed "<<kb<<endl;
cout<<" Press any key to clear screen";
getch();
clrscr();
cout<<" Press any key to Exit";
getch();
return 0;
}
```

یک مثال محاسباتی نکته دار
می خواهیم یک عدد ارز کاربر بگیریم نصفش را محاسبه و چاپ کنیم

```
#include<iostream.h>
#include<conio.h>
int main(){
int a;
float b;
cin>>a;
b=a/2;
cout<<b;
getch();
return 0;
}
```

حالا اگه کاربر یک عدد فرد مثلا سه وارد کند خروجی برنامه یک خواهد بود
بدلیل این که دو عدد صحیح بر یکدیگر تقسیم شده اند جواب نیز عددی صحیح
است در این مثال میبینید که حتی با تعریف متغیر b بعنوان ممیز شناور
مشکل همچنان پا بر جاست زیرا که حاصل تقسیم صحیح است و به نوع متغیر
مربوط نمی شود برای حل مشکل باید بنویسید `b=static_cast<float>(a/2);`
به این عمل تغییر نوع یا type casting میگویند که برای کامپایلر نوع عبارت
را تعریف میکند. من از این دستور استفاده کردم اما در توریو سی سه کار نکرد
بجای آن از دستور

`b=(float)a/2;` استفاده کردم که از دستور های زبان C است . این کامپایلر زیادی
قدیمی بهتر به بهتر شو گیر بیارید. مثلا Borland C++ 5.02 یا visual c++ 6 که البته
این مورد آخر دستور های مد گرافیک رو اجرا نمی کنه.