

به نام خداوند جان و خرد

دانشگاه فنی و حرفه ای دختران پروجرد (تزکیه)

آموزش مجازی درس برنامه سازی کامپیوتر

رشته سخت افزار

مقطع کاردانی

مدرس: شیرین احدی

زمستان 98

آشنایی با زبان ++C:

زبان ++C از زبان C ناشی شده و در آن ویژگی های جدیدی وجود دارد. در کل زبان های برنامه نویسی به سه دسته تقسیم میشود:

- 1- زبان سطح بالا
- 2- زبان سطح میانی
- 3- زبان سطح پایین

زبان های سطح بالا به زبان محاوره ای انسان نزدیکتر است و در زبان های سطح پایین میتوانیم به حافظه دسترسی داشته باشیم.

زبان ++C جز زبان های سطح میانی است.

++C یک زبان شی گراست که در برنامه های مختلف مورد استفاده قرار میگیرد. بدیت ترتیب سرعت تولید نرم افزار افزایش می یابد همچنین قابلیت خوانایی برنامه هایی که در این روش نوشته میشوند بالا بوده و عیب یابی آنها آسان است.

انواع داده:

برای برنامه نویسی نیاز به ورود داده ها و پردازش آنها و همچنین نیاز به استخراج نتایج داریم بنابراین باید نوع داده های ورودی را مشخص کنیم.

در زبان ++C شش نوع داده وجود دارد:

Int

Float

Double

Char

Bol

Void

:Int

این داده اعداد صحیح را ذخیره میکند و مقدار حافظه ای که برای آن مصرف میشود بستگی به سیستم عامل دارد. در سیستم های 16 بیتی دو بایتی است و در سیستم های 32 بیتی به صورت 4 بایتی است.

:Float

در این نوع داده اعداد اعشاری ذخیره میشود و میزان حافظه مصرفی آن 4 بایت است هنگامی که میخواهیم در حافظه صرفه جویی کنیم از این نوع داده استفاده میشود.

:Double

برای پذیرفتن اعداد اعشاری با دقت بالاتر استفاده میشود و میزان حافظه ی مصرفی آن 8 بایت است. البته محاسبات عددی روی double وقت گیر است.

بنابراین میتوان چنین گفت که هنگامی که نیاز به دقت بالاتری بود از نوع double و هنگامی که نیاز به سرعت بالاتری بود از float استفاده میکنیم.

:Char

برای ذخیره حروف و نمادها و حتی ارقام از آن استفاده میکنیم و هر آنچه بین ‘‘ قرار بگیرد به صورت کاراکتر میباشد. میزان حافظه ی مصرفی آن 1 بایت است.

: Bool

برای انجام عملیات منطقی میتوانیم از نوع داده های بول استفاده کنیم که میتواند دو مقدار True و False را بپذیرد. مقدار True در حافظه ی سیستم به صورت 1 ذخیره میشود و مقدار False در حافظه با 0 ذخیره میشود. و میزان حافظه ی آن 1 بایت میباشد.

:Void

به این نوع داده هیچ فضایی اختصاص داده نمیشود. بنابراین هیچ مقداری نمیپذیرد و تهی است. کاربرد آن در توابع و اشاره گر هادر نظر گرفته میشود.

تعریف متغیر:

مقدار نامی که برای خانه های حافظه در نظر گرفته میشود که الگوی آن به شکل زیر است:

نام نوع داده

مثال:

Int x;

Int x,y,z;

Float a;

Char ch;

تعریف ثوابت:

اگر بخواهیم مقداری را در حافظه ذخیره کنیم و این مقدار قابل تغییر نباشد از ثوابت استفاده میکنیم و نحوه ی تعریف آن به شکل زیر است:

مقدار=نام نوع داده Const

مثال:

Const float PI=3.14;

ساختار برنامه ++C:

قالب برنامه یک تابع به نام main است اگر نوع برگشتی آن void باشد دیگر لازم نیست در انتهای برنامه مقداری به آن برگردانده شود. اگر بدون void باشد return را در انتهای برنامه بنویسیم.

حالت اول:

```
Void main(){  
    ;اعلان متغیر  
    ;دستورات اجرایی  
}
```

حالت دوم:

```
Int main(){  
    ;اعلان متغیر  
    ;دستورات اجرایی  
    return 0;  
}
```

نکته: از تابع getch در انتهای برنامه به این منظور استفاده میشود که خروجی برنامه متوقف شود. تا کاربر بتواند نتیجه را مشاهده کند و به محض فشردن یک کلید از صفحه کلید اجرای برنامه را خاتمه دهد. تابع getch در فایل سرآیند conio.h تعریف شده است.

فایل سرآیند:

هر گاه بخواهیم از توابع و یا اشیا از پیش تعریف شده در برنامه استفاده کنیم باید فایل سرآیند(هدر فایل) را با استفاده از دستورات پیش پردازنده به ابتدای برنامه اضافه کنیم.

دستورات پیش پردازنده به صورت زیر تعریف میشود:

```
#include<....>
```

نکته: دستورات ورودی و خروجی در فایل سرآیند <iostream.h> قرار میگیرد.

بنابراین در کل میتوان چنین بیان کرد که ساختار های برنامه نویسی به دو صورت ذیل قابل نوشتن میباشد:

حالت اول:

```
#include<iostream.h>
#include<conio.h>
void main()
{
....
getch();
}
```

حالت دوم:

```
#include<iostream.h>
#include<conio.h>
Int main()
{
...
getch();
return 0;
}
```

نکته: در ساختار دوم برای اینکه سیستم عامل متوجه شود برنامه به درستی اجرا شده باید در انتها یک مقدار صحیح به آن برگردانده شود بنابراین از دستور `return 0` استفاده میکنیم.

دستورات ورودی و خروجی:

از دستور `cin` به همراه عملگر `>>` برای دریافت مقادیر هنگام اجرا از کاربران استفاده میکنیم. `cin` در فایل سرآیند `<iostream.h>` قرار دارد.

`Cout <<` نام متغیر

نکته: برای نوشتن اطلاعات در خروجی از دستور cout به همراه عملگر خروجی << استفاده میشود. دستور cout در فایل سرایند <iostream.h> قرار دارد.

cout<<متغیر<<

روش های مقدار دهی متغیر:

1- روش مستقیم: در این روش به محض تعریف متغیر، آن را مقدار دهی میکنیم

مثال:

```
int x=5;
```

```
int x=5 , y=10, z=15;
```

2- روش غیر مستقیم: ابتدا متغیر تعریف میشود و سپس هر جای برنامه که لازم بود با استفاده از دستور انتساب آنرا مقدار دهی میکنیم.

مثال:

```
int x;
```

```
...
```

```
x=5;
```

3- با استفاده از دستور ورودی: که در زمان اجرا به متغیر مقدار وارد میکنیم.

مثال:

```
int x;
```

```
cin >>x;
```