کارگاه کامپیوتر

جلسه : **دوم**

مدرس : **الهام یوسفی**

آشنایی با زبان C++

دستورات ورودی / خروجی

دستور خروجی

- این دستور برای نمایش اطلاعات (پیام، محاسبه، مقدار یا متغیر) استفاده می‌شود :

cout << "پیام" ;

cout << نام متغیر ;

- می‌توان همزمان پیام و متغیر را باهم چاپ کرد:

cout << "پیام" << نام متغیر/ مقدار / عمل محاسباتی ;

- در دستور خروجی می‌توان با برخی دستورات کنترلی، محل مکان نما و نمایش مقادیر را تا حدودی کنترل نمود.

توضیح	دستور
متن خروجی را به خط بعد می‌برد	\n یا endl
مکان نما را به اندازه یک tab (۸ کاراکتر) به جلو می‌برد	\t
چاپ \	\\
چاپ ' یا "	\' یا \"

انواع داده ها

به طور کلی انواع داده ها بصورت زیر است :

- **اصلی** : داده های استاندارد از قبیل int ، char ، float ، double
- **مشتق شده** : آرایه ، اشاره‌گر ، توابع و ...

انواع داده های صحیح

نوع	سایز بر حسب بایت	محدوده
int	2 یا 4	-2147843648 to 2147843647 یا -32768 to 32767
unsigned int	2 یا 4	0 to 2147843647 یا 0 to 32767
short	2	-32768 to 32767
unsigned short	2	0 to 65535
long	4	-2147843648 to 2147843647
unsigned long	4	0 to 4294967295

- **signed :** از آن برای علامت دار کردن یک نوع استفاده می‌گردد.
- **unsigned** : از این کلمه کلیدی برای بدون علامت کردن متغیرها استفاده می‌شود. متغیری که بدون علامت باشد قابلیت ذخیره کردن اعداد منفی را ندارد.
- **short** : این کلمه کلیدی ظرفیت متغیر خود را به نصف کاهش می دهد.(البته در برخی موارد ممکن است در ظرفیت متغیر خود تغییری ایجاد ننماید که این بستگی به نوع کامپایلر و ماشینی که برنامه بر روی آن کامپایل می‌شود دارد و از سیستمی به سیستم دیگر فرق دارد.)
- **long** : این کلمه کلیدی ظرفیت متغیر خود را بیشتر می نماید.

نکته: کلمات کلیدی signed و unsigned هیچ تغییری بر روی int در حجم اشغالی ندارند.

انواع داده های اعشاری

نوع	سایز بر حسب بایت	محدوده
float	4	3.4*10^-38 to 3.4*10^38
double	8	1.7*10^-308 to 1.7*10^308
long double	10,12	3.4*10^-4932 to 3.4*10^4932

- لازم به توضیح است که متغیرها در سخت افزارهای ۱۶بیتی، ۳۲بیتی و ۶۴ بیتی باهم متفاوت هستند در اینجا متغیرها در سخت افزار ۳۲ بیتی نوشته شده است.

داده های کاراکتری

- متغیرهایی که از نوع کاراکتری تعریف می‌شوند، می‌توانند هر یک از حروف الفبا و یا ارقام عددی یا علائم (... , & , @ , [, # , %) را بپذیرند.
- نوع کاراکتری زیرمجموعه نوع صحیح قرار می‌گیرد، زیرا هر کاراکتر بر روی صفحه کلید، توسط یک کد عددی صحیح به نام کد اسکی شناسایی می‌شود.

نوع	سایز بر حسب بایت	محدوده
char	1	-128 to 127
unsigned char	1	0 to 255
signed char	1	-128 to 127

سایر انواع داده

- **داده های منطقی :**
  - نوع این داده ها bool می‌باشد.
  - دارای دو مقدار true یا false می‌باشند.
  - کاربرد آنها در حلقه ها و ساختارهای تصمیم گیری می‌باشد.

- **void**
  - void به معنای "هیچ" یا "بدون نوع" است.
  - به عنوان مثال تابعی که هیچ مقداری برنمی‌گرداند، نوع بازگشتی آن باید void باشد. یا تابع بدون پارامتر می‌تواند void را بپذیرد.
  - نمی توان متغیری با نوع void تعریف کنید.

تعریف متغیر

برای تعریف یک متغیر در زبان C داریم :

Data_Type Variable_Name [= value] ;

نوع داده نام متغیر [= مقداردهی اولیه] ;

مثال :

```
int     x, y=12  ;

char  ch='y'  ;

bool  T=true;
```

عملوند و عملگر

- **عملوند (Operand)** : یک عملوند هدف یک عملیات ریاضی است. هر عبارت که بین دو عملگر قرار گیرد یا بعد از یک عملگر بیاید یک عملوند محسوب می‌گردد.

- **عملگرها (Operators)** : نمادهایی هستند که اعمال خاصی را انجام می‌دهند. عبارتند از :
  - عملگر انتساب
  - عملگرهای محاسباتی
  - عملگرهای رابطه ای
  - عملگرهای منطقی
  - عملگرهای ترکیبی
  - عملگر شرطی
  - عملگر sizeof
  - عملگر کاما

عملگرِ انتساب

از این عملگر (=) برای انتساب مقادیر به متغیرها یا انتساب متغیری به متغیر دیگر استفاده می‌شود.

```
int   i , a;
i = 10;
a= i+1;
```

عملگرهای محاسباتی

این عملگرها به دو دسته تقسیم می‌شوند :

- **عملگرهای باینری :** دارای دو عملوند هستند :
  - * (ضرب)
  - / (تقسیم) : اگر هر دو عملوند صحیح باشند، نتیجه تقسیم نیز صحیح خواهد بود.
  - % (باقیمانده تقسیم)
  - + (جمع)
  - - (تفریق)
- **عملگرهای یکتایی / یکانی :** دارای یک عملوند هستند:
  - ++ (عملگر افزایشی) : یک واحد به عملوند خود اضافه می‌کند.
  - -- (عملگر کاهشی) : یک واحد از عملوند خود کم می‌کند.
  - - یا + (علامت عدد)
  - عملگر sizeof()

مثال

```
int  a,b,c;
a=7;
b=3;
c=a/b;        //   c=2
```

```
int  a,b;
float  c;
a=7;
b=3;
c=a/b;        //  c=2.000000
```

```
float a,b,c;
a=7;
b=3;
c=a/b;        //  c= 2.333333
```

```
int   a;
float b,c;
a=7;
b=3;
c=a/b;        //  c= 2.333333
```

- int / int ⇨ int
- int / float ⇨ float
- float / int ⇨ float
- float / float ⇨ float

تابع FLOAT

- تابع float برای تبدیل اعداد صحیح به اعشاری می‌باشد.

```
int  a,b,c;
a=7;
b=3;
c=a/b;
cout << c ;          //  2
cout <<  (float) a/b ;    //  2.33333
```

عملگر افزایشی/کاهشی

- **++ بعد از متغیر** : یعنی ابتدا مقدار متغیر در پردازش شرکت کند سپس یک واحد به آن اضافه شود.

```
int   x=2 , y=4 , z;
z=x++ * y;          // z= 2*4 = 8   ,   x=2+1=3
```

- **++ قبل از متغیر** : یعنی ابتدا به مقدار متغیر یک واحد به آن اضافه شود سپس در پردازش شرکت کند.

```
int   x=2 , y=4 , z;
z=++x * y;          // x=2+1=3   ,  z= 3*4=12
```

- **-- بعد از متغیر** : یعنی ابتدا مقدار متغیر در پردازش شرکت کند سپس یک واحد از آن کم شود.

```
int   x=2 , y=4 , z;
z=x-- * y;          // z= 2*4 = 8   ,   x=2-1=1
```

- **-- قبل از متغیر** : یعنی ابتدا به مقدار متغیر یک واحد از آن کم شود سپس در پردازش شرکت کند.

```
int   x=2 , y=4 , z;
z=--x * y;          // x=2-1=1   ,  z= 1*4=4
```

اولویت بندی عملگرهای محاسباتی

در عبارات ریاضی عملگرهایی که دارای ارزش یکسانی هستند، پردازش از چپ به راست انجام می‌شود.

ردیف	اولویت	عملگر
1	()	پرانتز
2	sizeof	یکانی
3	++ ، --	افزایشی/کاهشی
4	-	تغییر علامت
5	* ، / ، %	ضرب، تقسیم، باقیمانده
6	+ ، -	جمع، تفریق

مثال :

3

4+3-6*2/5

1

2

4

عملگرهای رابطه ای (مقایسه ای)

این عملگرها برای مقایسه دو عملوند بکار می‌روند و نتیجه درست (1) یا نادرست (0) را برمی‌گرداند.

عملگر	نام	مثال	نتیجه
>	بزرگتر	3>3	0
>=	بزرگتر یا مساوی	3>=3	1
<	کوچکتر	2<5	1
<=	کوچکتر یا مساوی	5<=2	0
!=	مخالف یا نامساوی	3!=2	1
==	تساوی	2==3	0

عملگرهای منطقی

عملگرهای منطقی بر روی عبارات منطقی عمل می‌کنند.

- **عملگر && (AND منطقی)** : نتیجه زمانی درست است که هر دو عملوند مقدار درست داشته باشند، در غیر اینصورت نتیجه نادرست می‌دهد.
- **عملگر || (OR منطقی)** : نتیجه زمانی نادرست است که هر دو عملوند مقدار نادرست داشته باشند، در غیر اینصورت نتیجه درست می‌دهد.
- **عملگر ! (NOT منطقی)** : نتیجه درست را نادرست و نتیجه نادرست را به درست تبدیل می‌کند.

A	!A
0	1
1	0

A	B	||
0	0	0
0	1	1
1	0	1
1	1	1

A	B	&&
0	0	0
0	1	0
1	0	0
1	1	1

- اولویت بندی عملگرهای منطقی از بالا به پایین :
  - !
  - &&
  - ||

عملگرهای ترکیبی

ترکیبی از عملگرهای محاسباتی و = (انتساب) هستند.

عملگر	روش استفاده	عملکرد	i	نتیجه
+=	i+=2	i=i+2	2	4
-=	i-=2	i=i-2	4	2
*=	i*=2	i=i*2	2	4
/=	i/=2	i=i/2	4	2
%=	i%=2	i=i%2	2	0

عملگر کاما

- برای انجام چند عمل در یک دستور به کار می‌رود.

(عبارت ۲ , عبارت۱) = متغیر

در اینجا عملگر کاما موجب می‌گردد که ابتدا عبارت ۱ و سپس عبارت ۲ ارزیابی شود و نتیجه ارزیابی عبارت ۲، به متغیر مورد نظر نسبت داده شود. در این گونه موارد معمولاً عبارت ۱ و عبارت ۲ با یکدیگر مرتبط‌اند.

مثال:

```
a = (b=5 , b+15) ;
```

در عبارت مزبور، ابتدا b برابر ۵ قرار داده می‌شود و سپس عبارت b+15 محاسبه می‌گردد که نتیجه آن برابر ۲۰ خواهد بود. در پایان، این مقدار به متغیر a نسبت داده می‌شود، یعنی پس از اجرای دستور مزبور مقدار a برابر ۲۰ خواهد شد.

- کاربرد دیگری از عملگر کاما در دستور for (حلقه) است.

عملگر ؟

- عملگر ؟ با تست یک شرط، مقداری را نشان می‌دهد. می‌توان نتیجه را به یک متغیر هم نسبت داد. این عملگر بصورت زیر استفاده می‌شود:

متغیر = exp1 ? exp2 : exp3

ابتدا exp1 ارزیابی می‌شود، درصورتیکه نتیجه ارزیابی آن true شود، مقدار exp2 در غیراینصورت مقدار exp3 در متغیر قرار می‌گیرد.

- این عملگر معادل دستورات زیر است:

```
if (exp1) then
    متغیر = exp2
else
    متغیر = exp3
```

مثال :

```
x=10;
y= x>9 ? 100 : 200          //y= 100
```

تقدم عملگرها

اولویت	عملگر
۱	محاسباتی
۲	رابطه ای
۳	منطقی
۴	شرطی؟
۵	ترکیبی
۶	کاما

تعریف ثابت

مقادیر متغیرها می‌توانند در طول اجرا تغییر کنند. اما گاهی لازم است مقادیری داشته باشیم که در طول برنامه تغییر نکنند مانند عدد p در محاسبات ریاضی. در اینصورت ثابت ها بصورت زیر باید تعریف شوند :

 # define نام مقدار

const نوع نام= مقدار

- در استفاده از پیش پردازنده define عملاً متغیری تعریف نمی‌شود بلکه فقط یک جایگذاری ساده انجام می‌شود.
- در const یک متغیر است که هم آدرس دارد و هم دارای همه ویژگی های متغیر با این تفاوت که ایستا است.

مثال :

 # define P 3.14

const float Pi =3.14

دستور ورودی

- با استفاده از این دستور، داده ها از صفحه کلید دریافت شده و در متغیر ذخیره می‌شود.

cin >> نام متغیر ;

- درصورتیکه بخواهیم با یک دستور، بیشتر از یک متغیر دریافت نماییم باید بصورت زیر عمل نمود:

cin >> variable1 >> variable2 >> … >> variable n **;**

مثال

1. برنامه ای بنویسید که به متغیر a مقدار ۱۰۰ و به متغیر s مقدار C++ را نسبت دهد، یک ثابت هم تعریف شود، سپس آن ها را نمایش دهد.
2. برنامه قبل را طوری تغییر دهید که مقدار متغیرها با پیامی مناسب از کاربر دریافت شود.
3. برنامه ای نویسید که دو عدد دریافت کند، اگر دو عدد با هم برابر بودند در خروجی عبارت Equal چاپ شود در غیر اینصورت Not Equal (با استفاده از عملگر شرطی ؟)

تمرین

۱- برنامه ای بنویسید که ابتدا نام و نام خانوادگی را دریافت کرده سپس آنها را با فاصله یک tab از هم در بین گیومه نمایش دهد.

"Name"	"Family"

۲- نتیجه عبارت زیر را با پیامی مناسب در خروجی نمایش دهید:

(5>=3) && (3>100) || (4>3)

۳- برنامه ای بنویسید که دو عدد دریافت کند، با استفاده از عملگر شرطی (?) عدد بزرگتر را در خروجی چاپ کند.

۴- برنامه ای بنویسید که در آن یک ثابت (با استفاده از const یا پیش پردازنده define) تعریف کنید سپس عددی از ورودی دریافت نمایید اگر از مقدار ثابت بزرگتر بود عبارت"Yes" را چاپ کند در غیر اینصورت عبارت "No" (با استفاده از عملگر شرطی) .

۵- برنامه ای بنویسید که دو متغیر a و b را دریافت کرده، سپس مقادیر آن دو را جابه جا کند.