دانشگاه آزاد اسلامی، واحد مشهد
گروه مهندسی پزشکی

حسابگری الگوریتمی

مبحث سوم:

مبانی زبان C++

مهدی آذرنوش

http://azarnoosh.mshdiau.ac.ir

فهرست مطالب

- مبانی اولیه زبان C++
- کار با داده ها و اطلاعات
- آشنایی با عملگرها
- تبدیل انواع
- فرآیند آماده سازی و اجرای برنامه
- ساختار برنامه در C++

ویژگی های زبان C++

- C++ یک زبان میانی و ساخت یافته
- قابلیت انعطاف و قدرت بالای آن برای برنامه نویسی
- قابلیت برنامه نویسی سیستم
  - امکان بهره برداری از سخت افزار و نرم افزارها
- ارتباط بین C++ و زبان اسمبلی
  - امکان بهره برداری از قابلیت های اسمبلی در C++
- زبان کوچک
  - تعداد کلمات کلیدی آن ۶۲ کلمه است. (مانند نوع متغیرها یا دستورات خاص)
- حساسیت به حروف (Case Sensitive)
  - تمام کلمات کلیدی در این زبان با حروف کوچک نوشته می شود.
  - توصیه: تمام برنامه ها با حروف کوچک نوشته شود.

ویژگی دستورالعملهای زبان C++

- هر دستور به (;) ختم می شود.
- حداکثر طول یک دستور: ۲۵۵ کاراکتر
- دستور می تواند در چند سطر ادامه یابد.
- در هر سطر می توان چند دستور تایپ کرد.
  - این کار توصیه نمی شود.
- نوشتن توضیحات:
  - /* This is a sample comment */
  - // This is another sample comment

انواع داده ها

- char : کاراکتر ('a', 'b', 'x')
- int : اعداد صحیح (2, 125)
- float : اعداد اعشاری (15.5, 124.27)
  - (۷ رقم دقت)
- double : اعداد اعشاری بزرگتر از float
  - (۱۵ رقم دقت)
- bool : برای ذخیره مقادیر منطقی (دو ارزش true و false)
- void (تهی)

طول داده

- در پردازنده های مختلف متفاوت است:
  - **DOS و Win3.1 : ۱۶ بیتی**
  - **Other Windows : ۳۲ بیتی**

- ایجاد انواع جدید
  - Signed و Unsigned
  - Long و Short

طول داده - ۲

نوع داده	اندازه به بیت	بازهٔ مورد قبول
char	8	-128 _ 127
unsigned Char		0 _ 255
signed Char		-128 _ 127
int	16 or 32	-32768 _ 32767
unsigned int		0 _ 65535
signed int		-32768 _ 32767
short int	16	-32768 _ 32767
unsigned short int		0 _ 65535
signed short int		-32768 _ 32767
long int	32	-2147483648 _ 2147483647
unsigned long int		-2147483648 _ 2147483647
signed long int		0 _ 4294967295
float	32	دقت اعشار: ۷ رقم دقت
double	64	دقت اعشار: ۱۵ رقم دقت
long double	80	دقت اعشار: ۱۹ رقم دقت

تعریف متغیرها

- محل ذخیره داده ها در حافظه (نام محل حافظه)
- مقدار آنها در طول برنامه ممکن است تغییر کند.
- داده ها دارای نوع هستند پس متغیرها نیز باید دارای نوع باشند.
- تعیین بازهٔ اعداد مورد قبول و همچنین میزان حافظه مورد نیاز توسط نوع متغیر

نامگذاری متغیرها

- ترکیب حروف a تا z و A تا Z و ارقام و خط ربط (_)
- اولین کاراکتر رقم نباشد.
- طول آن هر مقداری می تواند داشته باشد ولی فقط ۳۱ کاراکتر اول نام متغیر استفاده می شود.
- از کلمات کلیدی نباشد.

اسامی مجاز	اسامی غیرمجاز
count	for
test24	24test
sum	grade!1
S_1	.pcx

کلمات کلیدی

- **کلمات کلیدی مشترک در C و C++**

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

- **کلمات کلیدی مختص C++**

asm	bool	catch	class	const_cast	delete
dynamic_cast	explicit	false	friend	inline	
mutable	namespace	new	operator		
private	protected	public	reinterpret_cast	static_cast	
template	this	throw	true	try	typeid	typename
using	virtual	wchar_t			

اعلان متغیرها

- تعیین نوع متغیر
  - قبل از بکارگیری هر متغیر آن را باید اعلان کرد.
  - اعلان متغیرها در هر جای برنامه امکان پذیر است.
    - توصیه می شود این کار در ابتدای برنامه صورت گیرد.
  - برای اعلان بیش از یک متغیر آنها را با کاما (,) از هم جدا می کنیم.
  - نوع متغیر را وابسته به نیاز تعریف می کنیم.

اعلان متغیرها- ۲

- فرمت اعلان متغیرها:

- نوع داده نام متغیر;

Example:

```
int x,y;
float m, n;
char ch1,ch2;
double d1;
long int p1;
bool b;
```

مقداردهی به متغیرها

- سه روش:
  - هنگام اعلان متغیر
  - با دستور انتساب (=)
  - دستورات ورودی

Example:

```
int x, y=5;
char ch1='a', ch2='m';
bool b1=false, b2=true;
```

Example:

```
float f1, f2;
char ch1, ch2;
f1=15.5;
f2=20.25;
ch1=ch2='a';
```

Example:

```
int x, y;
cin >>x >>y;
```

ثوابت

- مقادیری که در برنامه وجود دارند ولی قابل تغییر نیستند.
- قوانین نامگذاری آن مشابه متغیرها
- پس از اعلان ثوابت در برنامه قابل تغییر نیستند.
- کاربرد آن: قابلیت برنامه نویسی به صورت پارامتری

اعلان ثوابت

- استفاده از دستور #define

- #define <نام ثابت> <مقدار>
  - مقدار ثابت نوع آن را تعیین می کند.
  - در انتهای آن علامت (;) قرار نمی گیرد.
  - از دستورات پیش پردازنده است (قبل از ترجمه بجای نام قرار می گیرد و در زمان اجرا وجود ندارد.)
  - برای تفکیک از متغیرها بهتر است نام آنها با حروف بزرگ مشخص شود.

- استفاده از دستور const

- const <نوع داده> <نام ثابت> = <مقدار>;

Example:

```
#define  M        100
#define  PI       3.14
```

Example:

```
const  int M=100, count=40;
const signed char x='a';
```

تعریف عملگرها

- نمادهایی که اعمال خاصی انجام می دهند.
- استفاده در هنگام نیاز به انجام عملیات بر روی داده ها
- عملوند: مقادیری که عملگرها بر روی آنها عمل می کنند.

- انواع عملگرها
  - عملگرهای محاسباتی
  - عملگرهای رابطه ای
  - عملگرهای منطقی
  - عملگرهای ترکیبی
  - عملگرهای بیتی
  - عملگرهای متفرقه

عملگرهای محاسباتی

عملگر	نام	مثال
–	تفریق و منهای یکانی	x-y یا -x
+	جمع	x+y
*	ضرب	X*y
/	تقسیم	x/y
%	باقیمانده تقسیم صحیح	x%y
––	کاهش (decrement)	x--یا --x
++	افزایش (increment)	x++ یا ++x

عملگرهای محاسباتی

- تفاوت X-- ,--X : اگر این عملگر در عبارت قبل از عملوند بیاید، ابتدا این عملگرها عمل کرده و سپس نتیجه در محاسبات شرکت داده می شود، در غیر اینصورت مقدار فعلی عملوند در عبارت استفاده شده و سپس عملگر بر روی آن تأثیر می گذارد:

Example:	
int x=10, y;	
y=x++;	**x=11, y=10**
y=++x;	**x=12, y=12**
y=(x++ + ++x);	**x=14, y=26**

تقدم عملگرهای محاسباتی

Example:
```
int m, x=6, y=10;
m=x + y / 2 * 3;
```

- حاصل عبارت روبرو چقدر است؟

((x+y)/(2*3))=(16/6)=2

(((x+y)/2)*3)=(16/2)*3=24

(((y/2)*3)+x)=((10/2)*3)+x=21

(((y/2)+x)*3)=((10/2)+6)*3=33

++ −−
− (منهای یکانی)
* / %
+ −

عملگرهای دارای تقدم یکسان، تقدم مکانی دارند

عملگرهای رابطه ای

- استفاده در دستورات شرطی برای مقایسهٔ دو مقدار

عملگر	نام	مثال
>	بزرگتر	x>y
>=	بزرگتر یا مساوی	x>=y
<	کوچکتر	x<y
<=	کوچکتر یا مساوی	x<=y
==	تساوی	x==y
!=	نامساوی	x!=y

عملگرهای منطقی

- عبارات منطقی دارای دو ارزش درستی و نادرستی هستند
- در زبان C++:
  - ارزش نادرستی: صفر (ثابت false)
  - ارزش درستی: مقادیر غیر صفر (ثابت true معادل ۱)

عملگر	نام	مثال
!	نقیض (not)	!x
&&	و (and)	x>y && m<p
||	یا (or)	x>y || m<p

Example:

```
bool x, y, m, p, q;
x=false;
y=true;
m=x && y;
p=x || y;
q= !x;
```

تقدم عملگرهای رابطه و منطقی

!
> >= < <=
== !=
&&
||

عملگرهای ترکیبی

- عمل محاسبات و انتساب با هم

عملگر	نام	مثال	معادل
+=	انتساب جمع	x+=y	x=x+y
-=	انتساب تفریق	x-=y	x=x-y
*=	انتساب ضرب	X*=y	x=x*y
/=	انتساب تقسیم	x/=y	x=x/y
%=	انتساب باقیمانده تقسیم	x%=y	x=x%y

عملگرهای بیتی

- ایجاد امکان استفاده از C++ در کاربردهای اسمبلی
- عمل بر روی مقادیر بیت، بایت و کلمه
- فقط بر روی مقادیر int و char عمل می کند
  - با float، double، void و bool و سایر انواع پیچیده بکار نمی رود.
- کاربرد: خواندن بایتهای وضعیت دستگاههایی مانند چاپگر، مودم و تست وضعیت آنها

عملگر	نام
&	و (AND)
|	یا (OR)
^	یای انحصاری (XOR)
~	نقیض (NOT)
>>	شیفت به راست
<<	شیفت به چپ

عملکرد عملگرهای بیتی

~x	x^y	x|y	x&y	y	x
1	0	0	0	0	0
1	1	1	0	1	0
0	1	1	0	0	1
0	0	1	1	1	1

Unsigned char x	مقدار دودویی x	مقدار دهدهی x
x=7;	00000111	7
x= x << 2	00011100	28
x= x << 4	11000000	192
x= x >> 1	01100000	96
x= x >> 3	00001100	12

عملکرد عملگرهای بیتی - ۲

نتیجه	عملگر	y	x
00000100	&	00110110	10000100
10110110	|	00110110	10000100
10110010	^	00110110	10000100
01111011	~	-	10000100

عملگرهای متفرقه

- متغیرها (نام مکانهای حافظه) دارای آدرس خاص می باشند.
- عملگرهای & و *
  - & : دسترسی به آدرس متغیر
  - * : دسترسی غیر مستقیم به حافظه

Example:

p = &x;	**آدرس x در p قرار می گیرد.**
***p =5;**	**جایی که آدرس آن در p است (همان x)، برابر با ۵ می شود.**
m= *p;	**محتویات جایی که آدرسش در p است (۵) در m قرار می گیرد.**

عملگرهای متفرقه- ۲

- عملگر ؟
- این عملگر عبارتی را ارزیابی می کند و بر اساس ارزش آن، نتیجهٔ عبارت دیگری را در متغیری قرار می دهد
- متغیر = <عبارت ۱> ? <عبارت ۲> : <عبارت ۳>;
- اگر ۱ درست باشد مقدار ۲ در متغیر قرار می گیرد و گرنه مقدار ۳

Example:

```
int x,y;
x=5;
y= x>5 ? x*2 : x*5;
```

عملگرهای متفرقه-۳

- عملگر کاما (,)
- انجام چند عمل در یک دستور بکار می رود:

- متغیر = (<عبارت ۱> , <عبارت ۲>);
- عبارت ۱ با ۲ در ارتباط است: اول ۱ ارزیابی شده نتیجهٔ آن در ۲ استفاده شده، حاصل در متغیر قرار می گیرد.

Example:

```
int x,y,m=7;
y = (x=3*m , x/2*5);
```

عملگرهای متفرقه – ۴

- **عملگر sizeof**
  - عملگر زمان ترجمه است و برای تعیین طول متغیر یا داده بر حسب بایت
- sizeof متغیر ;
- sizeof (نوع داده);

Example:

```
int x,y, m;
x = sizeof  y;
m = sizeof (float);
```

- **عملگر ()**
  - بالا بردن تقدم عملگر داخل خود
  - ارزیابی از داخلی ترین پرانتز

```
y = 4*2/(3+1)+(6+(7-2));
```

تقدم عملگرها در حالت کلی

()
! ~ ++ -- sizeof
* / %
+ -
<< >>
< <= > >=
== !=
&
^
|
&&
||
?
= += -= *= /= %=

تعریف

- انواع مختلف داده ها می توانند به یکدیگر تبدیل شوند

- انواع آن:
  - تبدیل در عبارات محاسباتی
  - تبدیل در احکام انتساب

- تبدیل انواع در عبارات محاسباتی:
  - قانون موجود تبدیل انواع کوچکتر به انواع بزرگتر

تبدیل انواع در عبارت محاسباتی

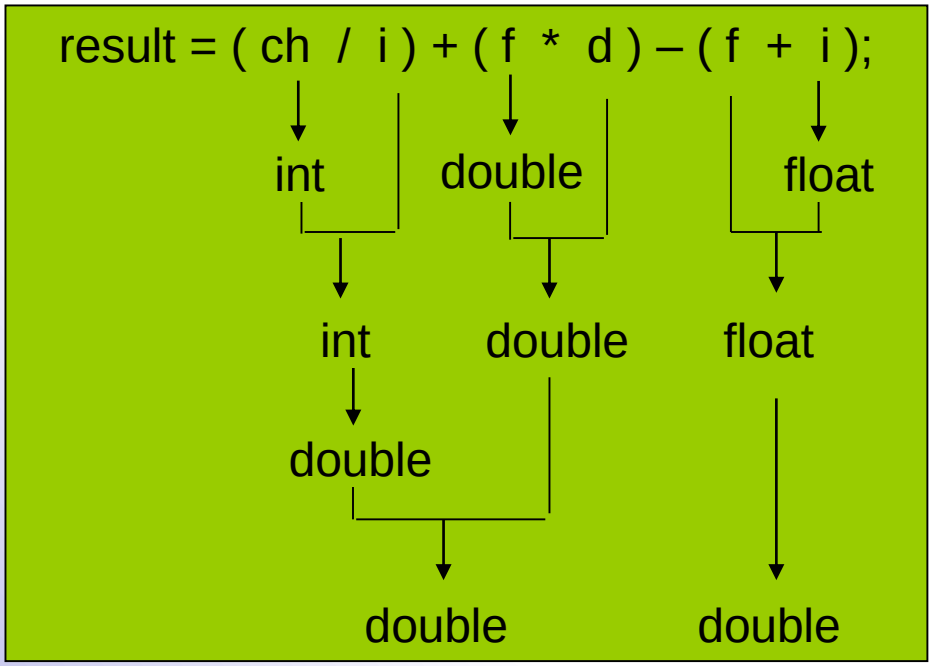
- اگر یکی از عملوندها **long double** باشد، عملوند دیگر به آن تبدیل می شود.
- وگرنه، اگر یکی از عملوندها **double** باشد، عملوند دیگر به آن تبدیل می شود.
- وگرنه، اگر یکی از عملوندها **float** باشد، عملوند دیگر به آن تبدیل می شود.
- وگرنه، اگر یکی از عملوندها **unsigned long** باشد، عملوند دیگر به آن تبدیل می شود.
- وگرنه، اگر یکی از عملوندها **long** باشد، عملوند دیگر به آن تبدیل می شود.
- وگرنه، اگر یکی از عملوندها **unsigned int** باشد، عملوند دیگر به آن تبدیل می شود.

- اگر یکی از عملوندها از نوع **Long** و عملوند دیگر از نوع **unsigned int** باشد ولی مقدار **unsigned int** را نتوان با **long** نمایش داد، هر دو به **unsigned long** تبدیل می شوند.

مثال تبدیل انواع

Example:

```
char ch;
int i;
float f;
double d;
result = (ch/i) + (f*d) – (f+i);
```



تبدیل انواع در احکام انتساب

- وقتی دو متغیر از انواع مختلف به یکدیگر نسبت داده شوند؛
- به عنوان مثال اگر متغیر نوع char به متغیر نوع int نسبت داده شوند.

Example:

```
char ch;
int x;
float f;
…
f=x;
f=ch;
x=f;
ch=x;
```

- انواع کاراکتری، اعشاری و دقت مضاعف به یکدیگر نسبت داده می شوند؛
- هر کدام از این انواع به دیگری قابل تبدیل است ولی باید دقت داشت که در تبدیل انواع بخشی از اطلاعات ممکن است از بین برود.

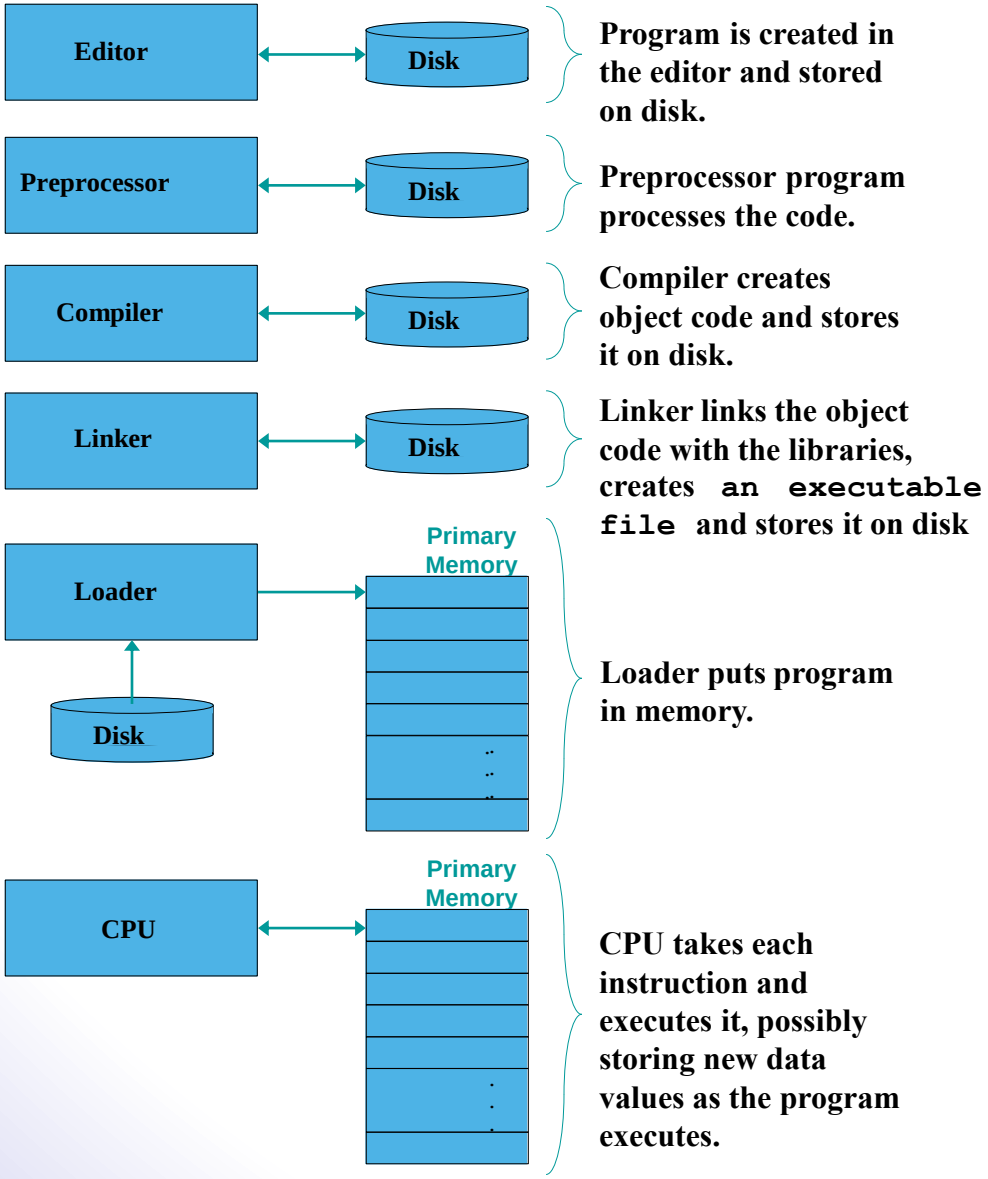
تبدیل انواع در احکام انتساب

- در انتساب یک مقدار اعشاری به یک مقدار صحیح، حداقل مقداری که از بین می رود، بخش اعشاری عدد است.

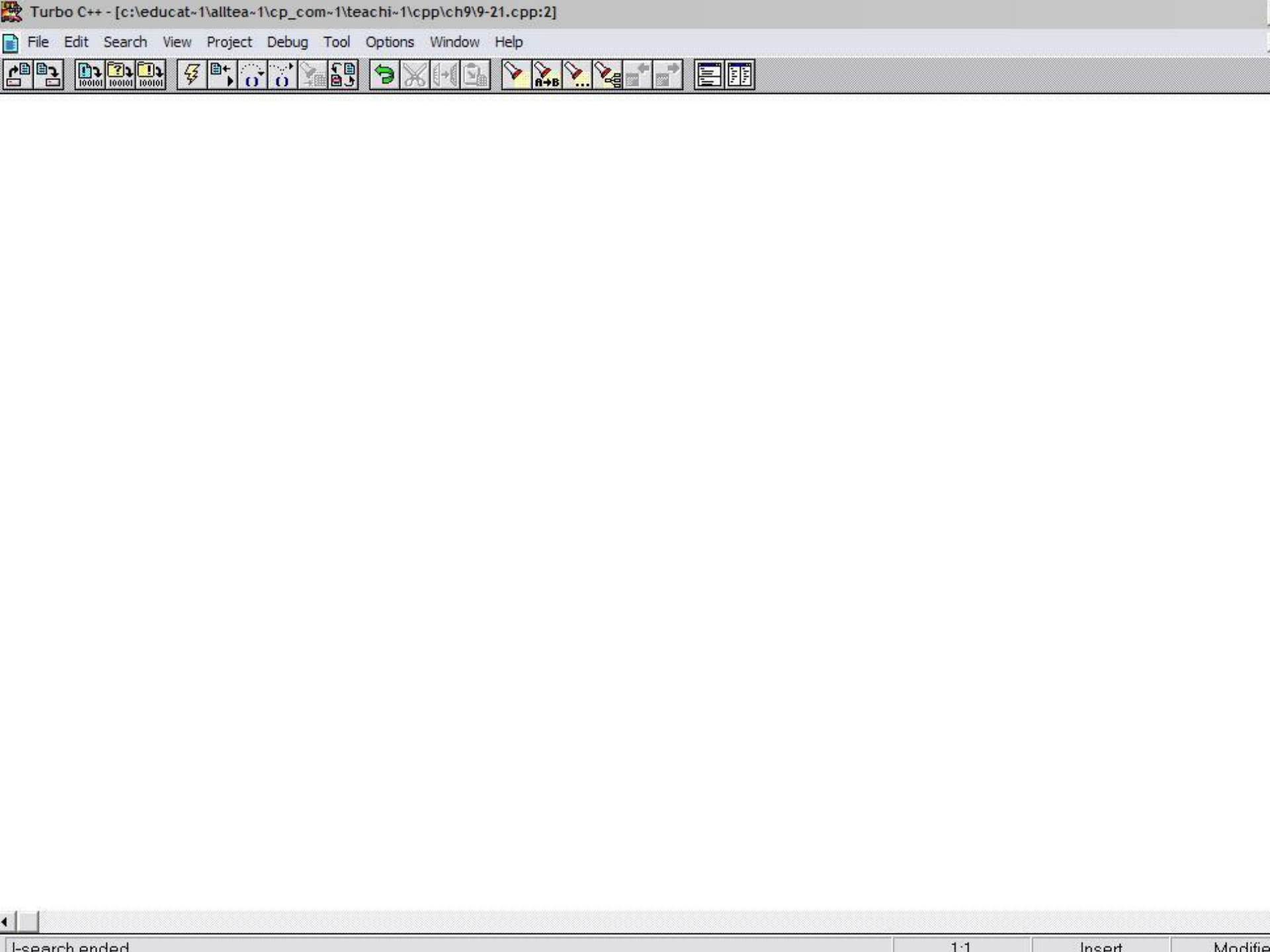
منبع	مقصد	اطلاعاتی که ممکن است از بین برود
char	signed char	اگر مقدار از ۱۲۷ بیشتر باشد مقصد منفی می شود
short int	char	۸ بیت با ارزش
int (16 bit)	char	۸ بیت با ارزش
int (32 bit)	char	۲۴ بیت با ارزش
long int	char	۲۴ بیت با ارزش
int (16 bit)	short int	اطلاعات از بین نمی رود
int (32 bit)	short int	۱۶ بیت با ارزش
long int	int (16 bit)	۱۶ بیت با ارزش
long int	int (32 bit)	اطلاعات از بین نمی رود
float	int	بخش کسری و یا بیشتر
double	float	دقت علائم کم می شود و نتیجه گرد می شود
long double	double	دقت علائم کم می شود و نتیجه گرد می شود

Phases of C++ Programs:

1. Edit
2. Preprocess
3. Compile
4. Link
5. Load
6. Execute



- وارد کردن برنامه در یک محیط ویراستاری
  - در کامپایلرهای C++ با پسوند cpp ذخیره می شود.
- ترجمهٔ برنامه
  - به منظور حذف اشکالات نحوی (Syntax error)
  - ایجاد فایلی با پسوند obj (به زبان ماشین و غیرقابل اجرا)
- پیوند زدن (link)
  - اتصال بخش های مختلف برنامه همراه با آدرس دهی کامل
  - ایجاد فایلی با پسوند exe که قابل اجرا است.


Turbo C++ - [c:\educat~1\alltea~1\cp_com~1\teachi~1\cpp\ch9\9-21.cpp:2]
File Edit Search View Project Debug Tool Options Window Help
I-search ended
1:1
Insert
Modifie

ساختار کلی زبان C++

- برنامه ها از تعدادی کلاس و تابع تشکیل می شود.
- هر تابع برای حل بخشی از مسأله نوشته می شود.
- نامگذاری توابع از قانون نامگذاری متغیرها پیروی می کند.

- انواع توابع:
  - توابع تعریفی توسط کاربر
  - توابع از پیش نوشته شده

تابع اصلی main()

- بدنهٔ اصلی برنامه، تابعی به نام main( ) است.
  - تمام برنامه های C++ باید دارای این تابع باشند. (از اجزای مهم زبان)
  - برای اطمینان از صحت عملکرد تابع main باید مقداری به سیستم عامل اجراکنندهٔ بازگردانده شود. (در برخی از کامپایلرها ضروری نیست)
  - مقدار بازگشتی به سیستم عامل باید از نوع int باشد لذا خود تابع main نیز باید به صورت int تعریف شود:
    - int main ( )
  - مقدار بازگشتی در انتهای برنامه با دستور return 0; داده می شود.
  - اگر بخواهیم سیستم مقداری را بازنگرداند:
    - void main( )
  - دستورالعملهای برنامه با { آغاز و با } پایان می یابد.

توابع از پیش نوشته

- نوشته شده همراه با کامپایلرها ارائه می شود
- بسیاری از اعمال در برنامه نویسی توسط این توابع انجام می شود.
- {مثلاً توابعی که به ورود داده ها از صفحه کلید و نوشتن آنها در صفحه نمایش مربوط است}
- توابع از قبل نوشته شده در فایلهای header قرار دارند.
- {پسوند فایلها .h و در شاخهٔ include قرار دارند}
- هر تابع مورد استفاده در فایل header خاصی قرار دارد که باید آن را در ابتدای برنامه اضافه کنیم.
- اتصال فایلهای header به برنامه از طریق دستور #include صورت می گیرد. {از دستورات پیش پردازنده}

نکات مهم در مورد فایل های سرآیند

- معمولاً قبل از تابع main( ) می آید.
- این دستور به ; ختم نمی شود.
- طریقهٔ نوشتن به صورت زیر است:

- #include <نام فایل header>

Example:
#include <iostream.h>

- بین # و include نباید فاصله ای باشد.
- ذکر علائم < و > ضروری است.
- بین نام فایل و علائم نباید فاصله ای باشد.
- برای اضافه کردن چند فایل header، هرکدام با یک دستور #include

مثالی از ساختار یک برنامه

```
1   // First_Example.cpp
2   // A first program in C++
3   #include <iostream>
4
5   // function mai
6   int main()
7   {
8    cout << "Welcome to C++!\n";
9
10  return 0;   // indi
11
12  } // end function main
```

Single-line comments.

Preprocessor directive to include input/output stream header file **<iostream>**.

Function **main** returns an integer value.

Left brace **{** begins function body.

program..

Statements end with a semicolon **;**.

Stream insertion operator.

Corresponding right brace **}** ends function body.

Keyword **return** is one of several means to exit function; value **0** indicates program terminated successfully.

```
Welcome to C++!
```

دستورات چاپی خروجی

- cout :برای چاپ اطلاعات در صفحه نمایش
- فایل سرآیند مربوطه: iostream.h

- فرمت:

- cout << عبارت ۱ >> عبارت ۲ >>... ;

- برای چاپ متن ها باید آنها را در کوتیشن (" ") قرار داد.
- متن می تواند حاوی کاراکتر کنترلی باشد.
  - شکل خروجی اطلاعات را مشخص می کنند.
  - کاراکترهای کنترلی با علامت \ شروع می شوند.

کاراکترهای کنترلی

کاراکتر	عمل متناظر
\n	انتقال کنترل به خط جدید (new line)
\t	انتقال به ۸ کاراکتر بعدی در صفحه نمایش (tab)
\a	بوق سیستم را به صدا در می آورد
\"	چاپ کوتیشن (")
\\	Back slash
\v	انتقال کنترل به ابتدای ۸ سطر بعد
\b	کاراکتر قبل از خودش را پاک می کند
\r	کلید Enter را مشخص می کند

مشاهده نتایج اجرای برنامه

- نمایش خروجی برنامه در صفحهٔ خروجی
- استفاده از getch( ) برای ماندن در صفحهٔ نمایش (برخی کامپایلرها)
- فایل سرآیند مربوطه: conio.h
- مثال ۳-۱: مشاهدهٔ عملکرد تابع getch() در برنامه ای که می خواهد عبارت "C++ is a good language" را چاپ کند.

مثال ۳-۱:

۱- عبارت C++ is a good language را چاپ کن.
۲- یک کاراکتر از ورودی دریافت کن.
۳- پایان

Example 3_1:

```
#include <iostream.h>
#include <conio.h>
int main()   {
    cout << " C++ is a good language.";
    getch();
    return 0;
}
```

```
C++ is a good language.
```

چاپ اطلاعات با cout

- مثال ۳-۲: چاپ اعداد ۱۰ و ۵/۱۵ در صفحه نمایش

مثال ۳-۲:

۱- 10 ⟵ x و 15.5 ⟵ y
۲- x و y را چاپ کن.
۳- پایان

Example 3_2:

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int x=10;
    float y=15.5;
    cout << " x = " << x;
    cout << " y = " << y;
    getch();
    return 0;
}
```

```
x = 10 y = 15.5
```

چاپ اطلاعات با cout – ۲

- مثال ۳-۳: چاپ اعداد ۱۰ و ۲۰ در دو سطر متوالی

مثال ۳-۳:

۱- 10 ⟵ x و 20 ⟵ y
۲- x و y را چاپ کن. (در دو سطر متوالی)
۳- پایان

Example 3_3:

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int x=10, y=20;
    cout << " x = " << x << "\n";
    cout << " y = " << y;
    getch();
    return 0;
}
```

```
x = 10
y = 20
```

چاپ اطلاعات با cout – ۳

- مثال ۳-۴: چاپ دو عدد صحیح ۱۰و ۲۰ و مجموع مربعات آنها

مثال ۳-۴:

۱- 10 ⟵ x و 20 ⟵ y
۲- x و y را چاپ کن.
۳- x*x+y*y را چاپ کن.
۴- پایان

Example 3_4:

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int x=10, y=20;
    cout << " x = " << x << ", y = " << y << endl;
    cout << " sum of squares is : " << x*x + y*y;
    getch();
    return 0;
}
```

```
x = 10 , y = 20
sum of squares is : 500
```

خواندن اطلاعات با cin

- cin :برای خواندن اطلاعات از صفحه کلید
- فایل سرآیند مربوطه: iostream.h

- فرمت:

- cin >> متغیر ۱ >> متغیر ۲ >>... ;

  - برای وارد کردن اطلاعات باید کلید Enter زده شود.
  - اگر چند داده باید با هم خوانده شود، باید بین هر دو داده حداقل یک فاصله باشد و در انتهای آخرین اطلاعات Enter زده شود.

خواندن اطلاعات با cin – ۲

- مثال ۳-۵: دریافت شعاع دایره (عدد صحیح) از ورودی و محاسبه و نمایش مساحت و محیط آن

Example 3_5:

```
#define  PI  3.14
#include <iostream.h>
#include <conio.h>
int main()
{
    int r;
    float area, p;
    cout << “Enter the radius : ”;
    cin >> r;
    area = PI * r *r;
    p = 2 * PI * r;
    cout << “area= “ << area << “perime= “ << p;
    getch();
    return 0;
}
```

مثال ۳-۵:

۱- r را بخوان.
۲- 3.14*r*r ← area
۳- 2*3.14*r ← p
۴- area و p را چاپ کن.
۵- پایان

```
Enter the radius : 5
area = 78.5 perime = 31.4
```

خواندن اطلاعات با cin – ۳

- مثال ۳-۶: محاسبه میانگین سه عدد صحیح که از ورودی خوانده شده

Example 3_6:

```
#include <iostream.h>
#include <conio.h>
int main()
{
    int x, y, m;
    float ave;
    cout << "Enter three numbers : ";
    cin >> x >> y >> m;
    ave = (float) (x+y+m) / 3;
    cout << "ave= " << ave;
    getch();
    return 0;
}
```

مثال ۳-۶:

۱- x و y و m را بخوان.
۲- (x+y+m)/3 ⟵ ave
۳- ave را چاپ کن.
۵- پایان

type casting

```
Enter the numbers : 5 20 18
ave = 14.33
```

کار با صفحه نمایش خروجی

- پاک کردن صفحه نمایش:
  - پاک کردن صفحهٔ نمایش برای مشاهده خروجی برنامه
  - استفاده از تابع clrscr()
  - فایل سرآیند مربوطه: conio.h

- انتقال مکان نما در صفحه نمایش
  - برای دریافت و یا چاپ اطلاعات در نقطه خاصی از صفحه نمایش
  - استفاده از تابع gotoxy(int x, int y)
  - x شمارهٔ ستون و y شمارهٔ سطر
  - فایل سرآیند مربوطه: conio.h

کار با صفحه نمایش خروجی - ۲

- مثال ۳-۷: محاسبه محیط و مساحت مستطیل با دریافت اطلاعات از ورودی و با پاک کردن صفحه نمایش و چاپ اطلاعات خروجی در مکان خاص

مثال ۳-۷:

۱- len و wid را بخوان.
۲- p ⟵ (len+wid)/2
۳- s ⟵ len*wid
۴- s و p را چاپ کن.
۵- پایان

```
Example 3_7:
#include <iostream.h>
#include <conio.h>
int main()
{
    int len, wid, p, s;
    clrscr();
    cout << "Enter length, width: ";
    cin >> len >> wid;
    p = (len+wid) *2;
    s = len * wid;
    clrscr();
    gotoxy(35,13);
    cout << "s = " << s << " p = " << p;
    getch();
    return 0;
}
```

```
s = 200 p = 60
```

خواندن کاراکترها از ورودی

- استفاده از تابع get() که عضو cin
  - علاوه بر cin
- فایل سرآیند مربوطه: iostream.h
- فرمت:

- ch = cin.get();

کار با صفحه نمایش خروجی – ۳

- مثال ۳-۸: نمایش یک کاراکتر که از ورودی خوانده شده است

مثال ۳-۸:

۱- ch را بخوان.
۲- ch را چاپ کن.
۳- پایان

```
Example 3_8:
#include <iostream.h>
#include <conio.h>
int main()
{
    char ch;
    clrscr();
    cout << “Enter a character: ”;
    ch = cin.get();
    clrscr();
    cout << “You typed character: “ << ch;
    getch();
    return 0;
}
```

```
You typed character: b
```