面向对象程序设计
——基于C++，谭浩强

主讲人：邵 平,18219074969
计算机学院 2018年3月

班级	人数	课 程 名	课程性质	学时分配				授课周数	考核类型	备注
				总学时	理论	实验	周学时			
16科技2	62	面向对象程序设计	必修	64	32	32	4	1-16周	考试	
16科技2	62	专业技能训练-面向对象程序设计	必修	32	0	32	32	17周	考查	

重要提示

- 上机，要带U盘，带上机指导书。
- 面向对象程序设计总分（考试）：考勤+作业30%，考试70%。3次（含3次）以上无故缺勤不能参加考试，无成绩。
- 专业技能训练总分（考查）：考勤+上机练习30%，考查70%。3次（含3次）以上无故缺勤不能参加考查，无成绩。

重要提示

- 答疑时间地点：周四晚上，中巴软件大楼105室（或发QQ邮件给我：617536919@qq.com）
- 课件下载：

ftp://172.21.85.11

用户名:16kj,密码:11111111

- 作业提交

ftp://172.21.85.11

用户名:16kjhomework,密码:11111111

学习方法

学习方法：

1、看除教材以外的参考书；建议加入程序员联合开发网；可自学与C++相关的MFC；软件要用原版(英文版)；上机教材中的GCC编译环境课外时间掌握，必须要学会用两种以上的编译器写*.CPP程序。

注：MFC(Microsoft Foundation Classes)，是一个微软公司提供的类库（class libraries），以C++类的形式封装了Windows的API函数，并且包含一个应用程序框架，以减少应用程序开发人员的工作量。
即使你不使用MFC框架，花点时间学习一下MFC的封装机制对你熟悉C++的OOP机制和Windows底层功能也是很有好处的。

附录（见书后的附录）

- ASCII码表
- 运算符优先级与结合性（见下页）：**从上到下优先级递减！**
- 口诀：**初单算移关，位逻条赋逗。**
  单条赋，右至左。

C语言中的运算符

说明	运算符	结合性
初等运算符	() [] -> .	->
单目运算符	! ~ ++ -- - (类型) * & sizeof	<-
算术运算符	* / %	->
算术运算符	+ -	->
移位运算符	<< >>	->
关系运算符	> >= < <=	->
关系运算符	== !=	->
按位与	&	->
按位异或	^	->
按位或	|	->
逻辑与	&&	->
逻辑或	||	->
条件运算符	?:	<-
赋值运算符	= += -= *= /= %= <<= >>= &= ^= |=	<-
逗号运算符	,	->

注:从上到下优先级递减!
总结:初单算移关,位逻条赋逗.
单条赋,右至左.

C++语言中的运算符

运算符	描述	结合性
::	二元作用域 一元作用域	->
() [] . -> ++ -- typeid dynamic_cast<type> static_cast<type> reinterpret_cast<type> const_cast<type>	初等/类型转换 ++--为一元后自增	->
++ -- + - ! ~ (类型) sizeof & * new new[] delete delete[]	++--为一元前自增	<-
.* ->*	取指向成员的指针	->
* / %	算术运算符	->
+ -	算术运算符	->
<< >>	移位运算符	->
< <= > >=	关系运算符	->
== !=	关系运算符	->
&	按位与	->
^	按位异或	->
|	按位或	->
&&	逻辑与	->
||	逻辑或	->
?:	条件运算符	<-
= += -= *= /= %= &= ^= |= <<= >>=	赋值运算符	<-
,	逗号运算符	->

第1讲 C++初步(1)：C++概述

C++是在C语言的基础上发展和完善的。而C语言有以下特点：

1、C语言吸收了其它语言的优点，实用性很强。

2、C语言是一种结构化的程序设计语言，简洁、使用灵活。

3、C既有高级语言的特点，又具有汇编语言的特点。

4、C程序的可移植性好。

5、C程序的语法结构不严密，程序设计的自由度大。

缺点：C语言对数据类型检查的机制比较弱；缺少支持代码重用的结构；随着软件工程规模的扩大，难以适应开发特大型的程序等等。

为了克服C语言本身存在的缺点，并保持C语言简洁、高效，与汇编语言接近的特点，1980年，贝尔实验室的Bjarne Stroustrup博士及其同事对C语言进行了改进和扩充，引入类的概念。并在1983年由Rick Maseitti提议正式命名为C++（C Plus Plus）。后来，又把运算符的重载、引用、虚函数等功能加入到C++中，使C++的功能日趋完善。

当前用得较为广泛的C++有：VC++ （Visual C Plus Plus）、 GNU G++、BC++（Borland C Plus Plus）、AT&T C++等。

贝尔实验室

美国贝尔实验室是晶体管、激光器、太阳能电池、发光二极管、数字交换机、通信卫星、电子数字计算机、蜂窝移动通信设备、长途电视传送、仿真语言、有声电影、立体声录音，以及通信网等许多重大发明的诞生地。自1925年以来，贝尔实验室共获得两万五千多项专利。一共获得8项诺贝尔奖（其中7项物理学奖，1项化学奖）。

Dennis Ritchie的设计原则：保持简单和直接（Keep it simple stupid，著名的KISS原则）

自由软件之父
理查德·斯托曼Richard Stallman

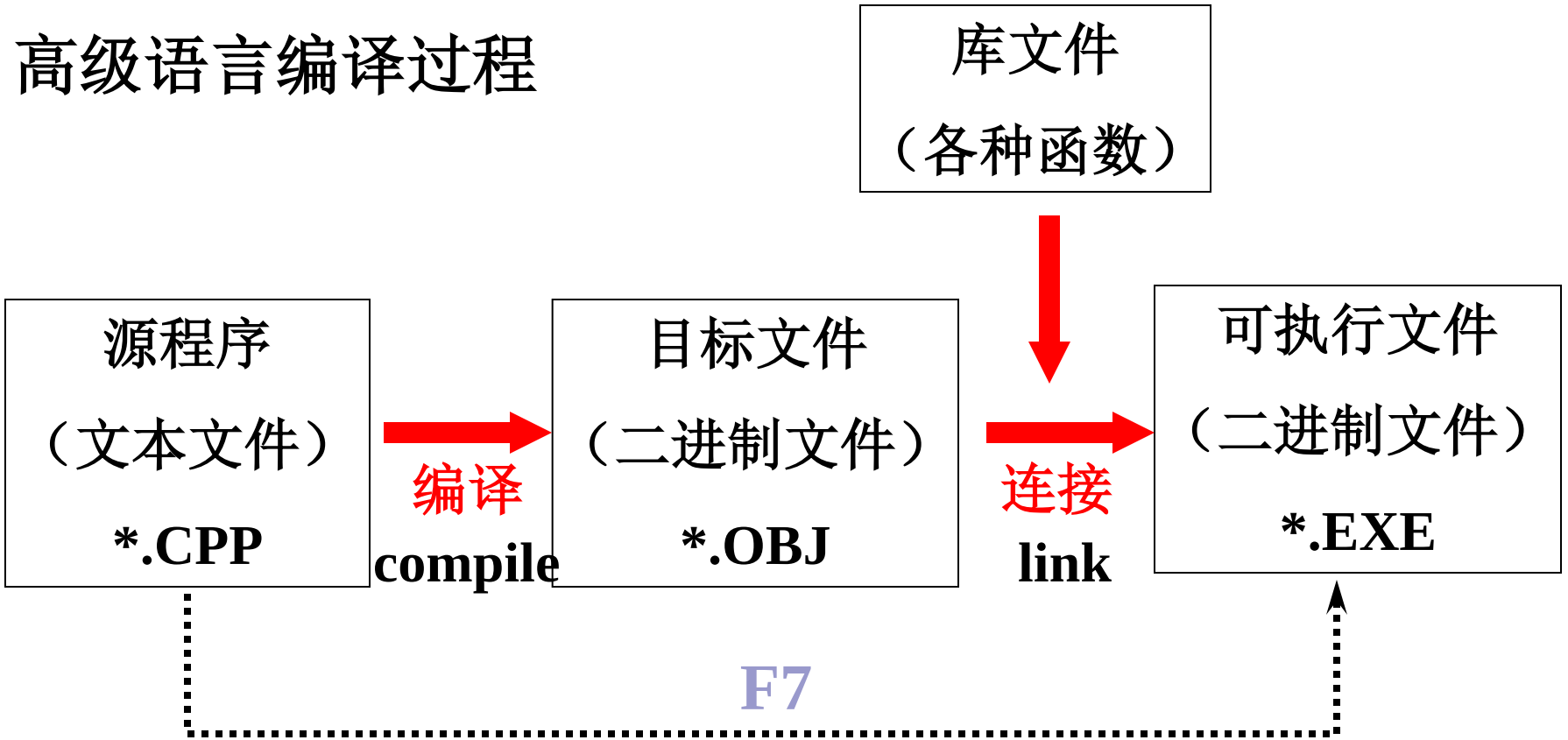
1985年3月，美国自由软件斗士Richard Stallman(RMS)发表了著名的**GNU宣言**，它的目标是创建一套完全自由的操作系统，到现在已有30周年的历史了。而他本人也因为宣言中的一句话而广为人知：**软件，既是用户控制着程序，也是程序控制着用户。**

一款软件要符合什么样的标准，才能算是“自由软件”？斯托曼给出了四个标准：用户能够自由运行软件；能够按照自己的意愿改写软件，并与他人合作，进行软件的再次开发；能够自由传播、分发软件；能够自由传播、分发软件的修改版本。

IT圈外不知道比尔·盖茨的人，恐怕很难成为富翁;而IT圈里，若有人不知道Richard Stallman，恐怕其“IT人”之名都难保了。盖茨虽然创建了前所未有的软件帝国，但在软件的发展史上，Richard Stallman的贡献更大，他是自由软件运动的领袖，是当之无愧的无冕之王!

简单的C++程序介绍

高级语言编译过程



在Visual C++系统中，可直接从源程序编译连接至可执行程序，但依然要生成*.OBJ及*.EXE。

例1：一个简单的C++程序

```
#include<iostream.h>
void  main(void )
{   cout<<“I am a student.\n”;  //输出字符串
}
```

包含文件

主函数

函数体开始

分号，一条完整语句的结束符

输出流，在屏幕上打印引号内的字符串

注释或说明

函数体结束

本程序编译执行后，在DOS屏幕上打印出

I am a student.

编译过程：

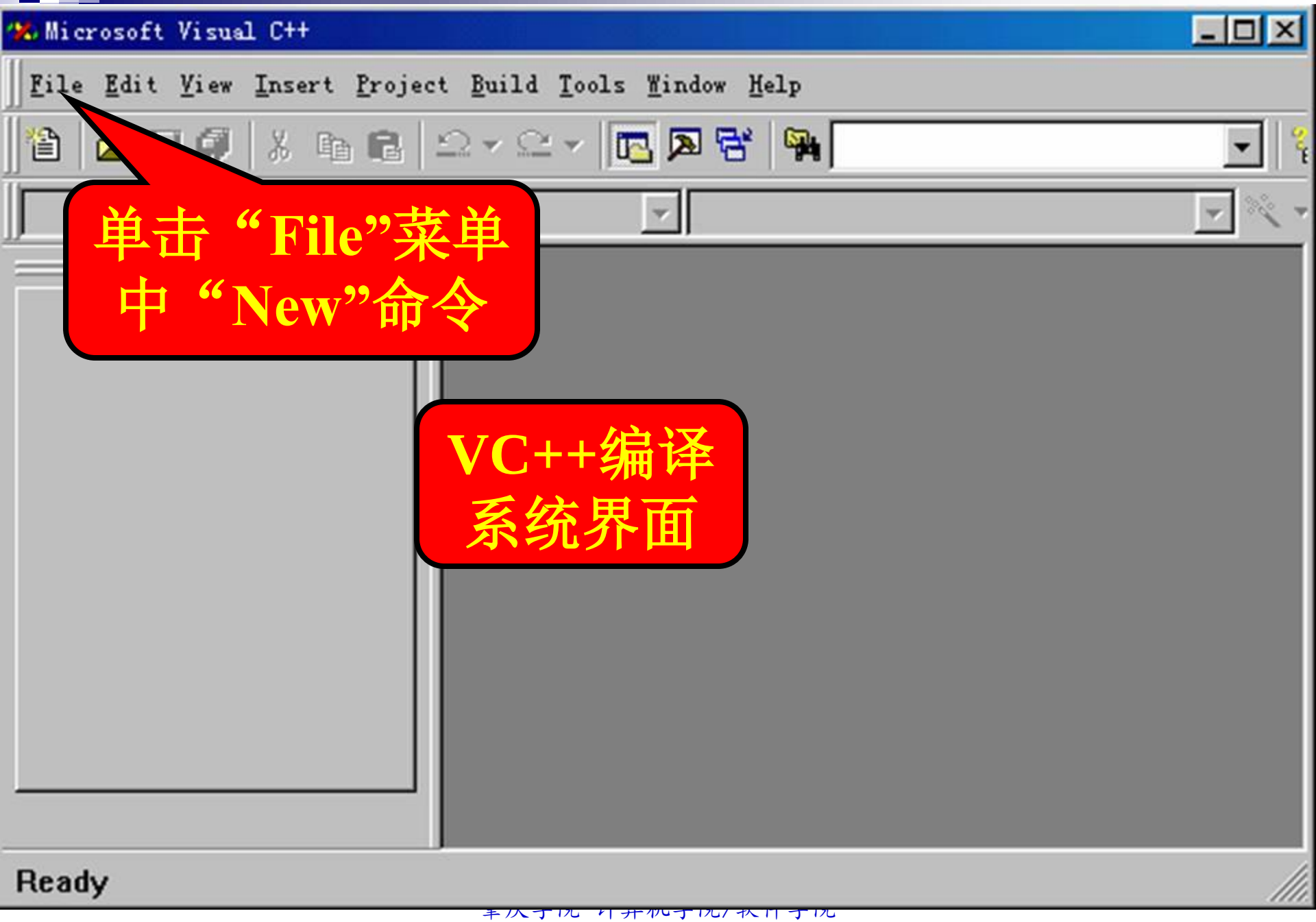
1）启动Visual C++,选择“文件”菜单中的“新建”命令，选择“文件”标签中的“C++ Source File”选项。

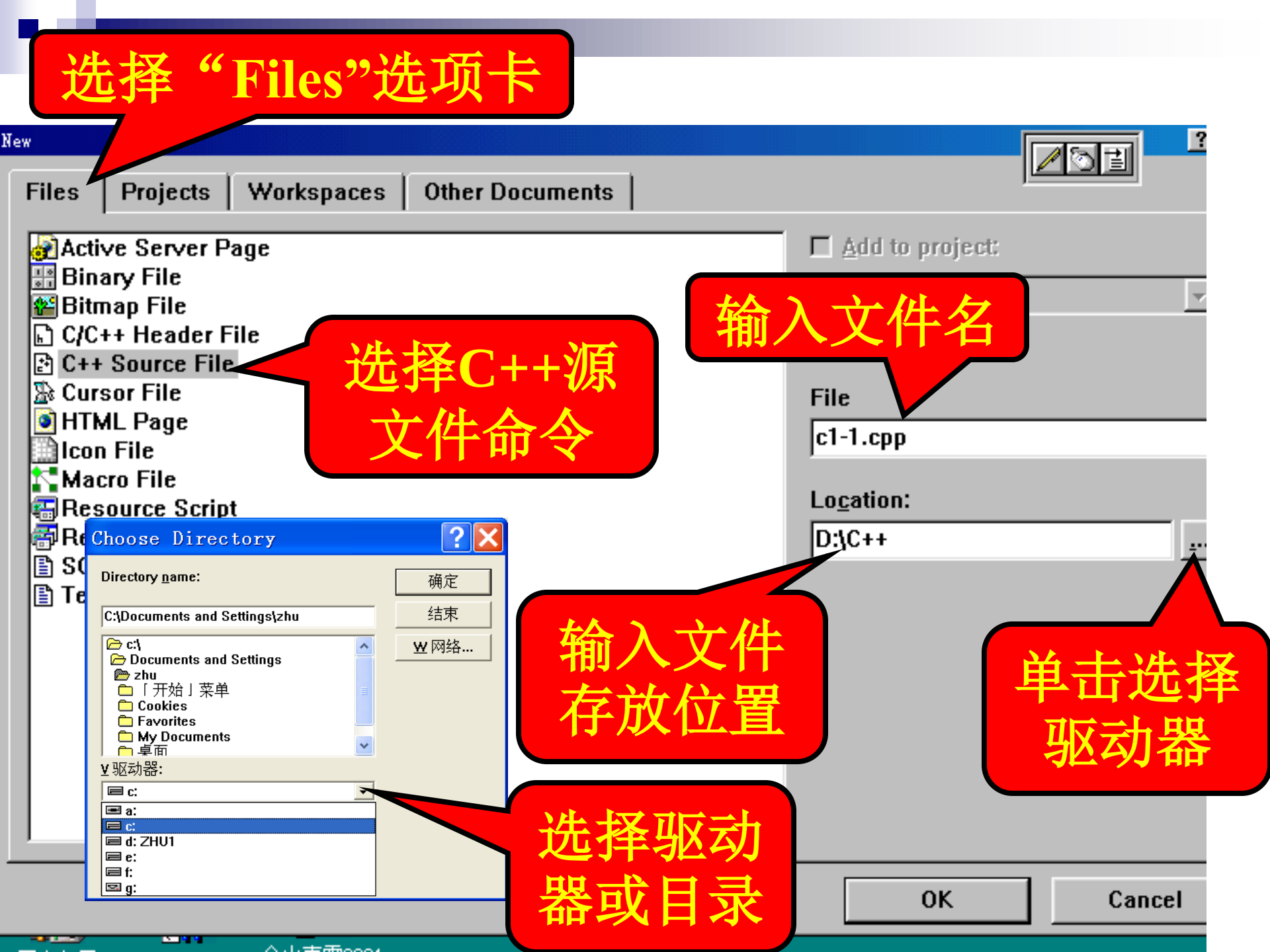
2）选择源程序存放的目录和输入源程序名，单击“确定”。

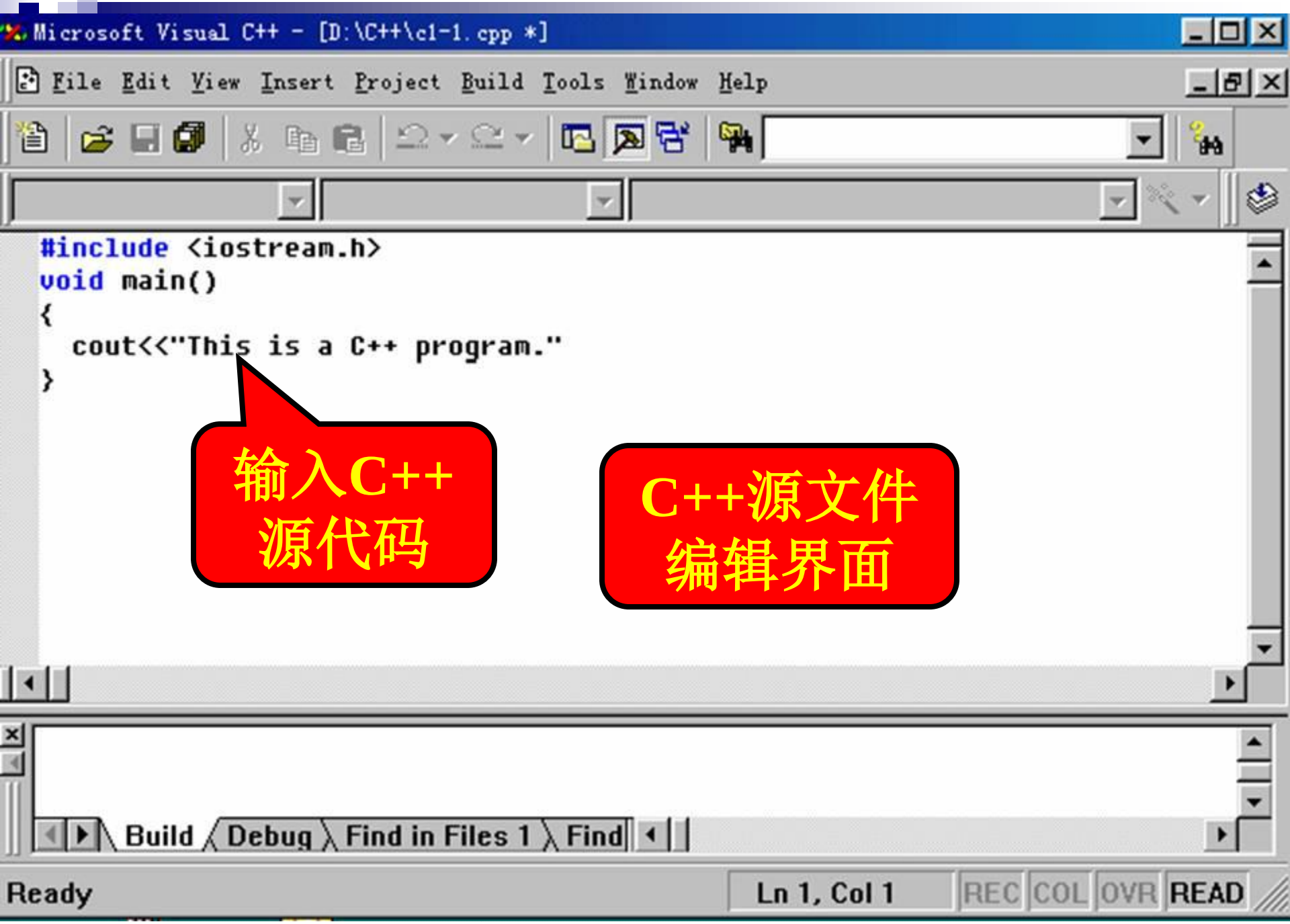
3）在编辑器中编写源程序。

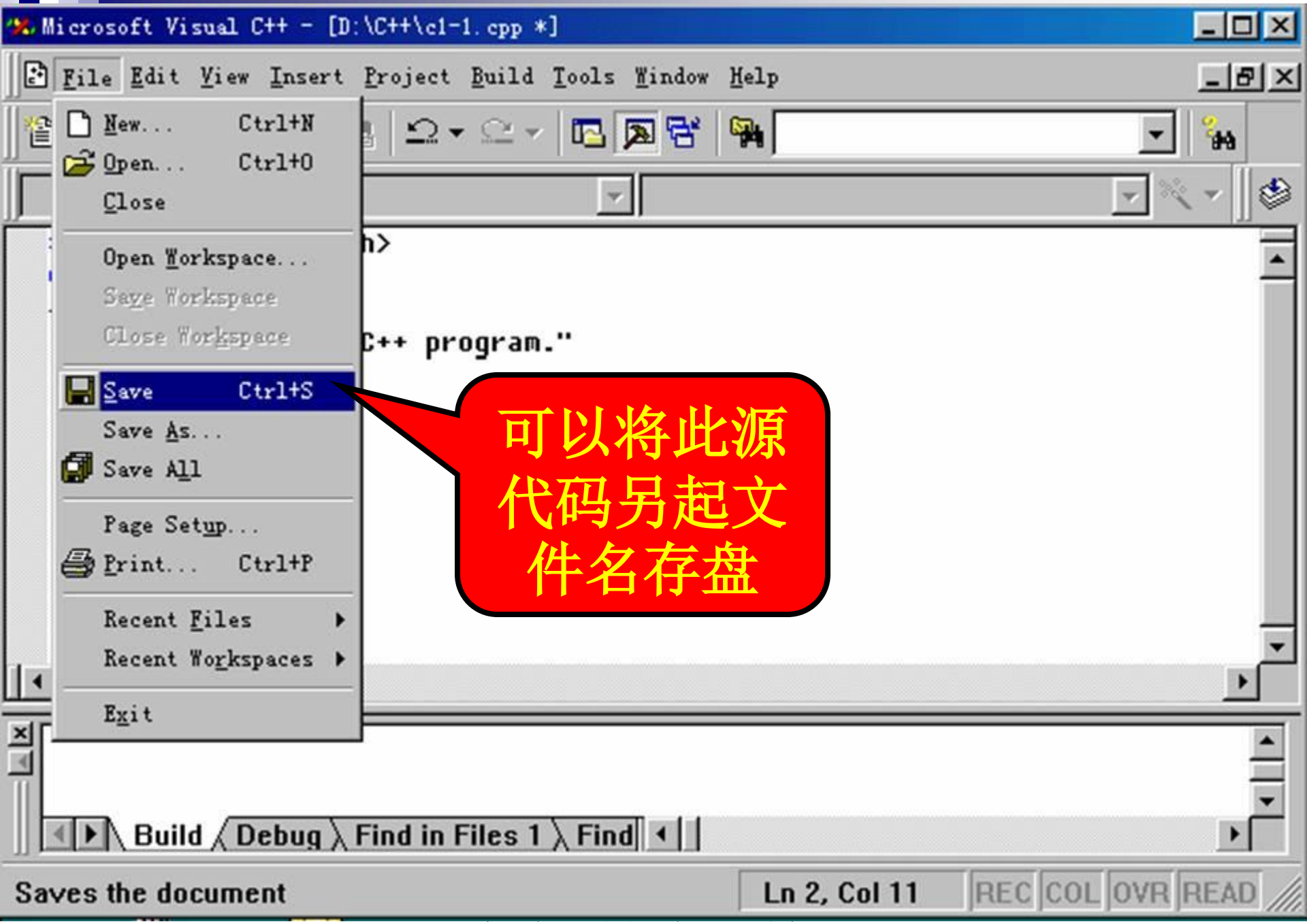
4）单击F7或“编译”中的“重建全部”编译源程序，若编译通过，单击“执行”，在DOS屏上看结果，任按一键返回编辑器。


我的文档
金山词霸 2002
WinPoET Dialer
我的电脑
网上邻居
陈鸿一鸣公文包
WinRAR
回收站
连接到 Internet
Internet Explorer
Outlook Express
Microsoft Visual Studio 6.0
Microsoft Visual C++ 6.0
Microsoft Excel
Microsoft Access
Microsoft FrontPage
Microsoft Outlook
Microsoft PowerPoint
Microsoft Word
Acrobat Distiller 5.0
Adobe Acrobat 5.0
Windows Update
WinRAR
打开 Office 文档
新建 Office 文档
洪恩奇境英语 (2)
程序(P)
文档(D)
设置(S)
搜索(C)
帮助(H)
运行(R)...
关机(U)...
Windows 2000 Professional
开始
修改VC++教材
Microsoft Po...
第1章 C++...
23:40
启动VC++
编译系统

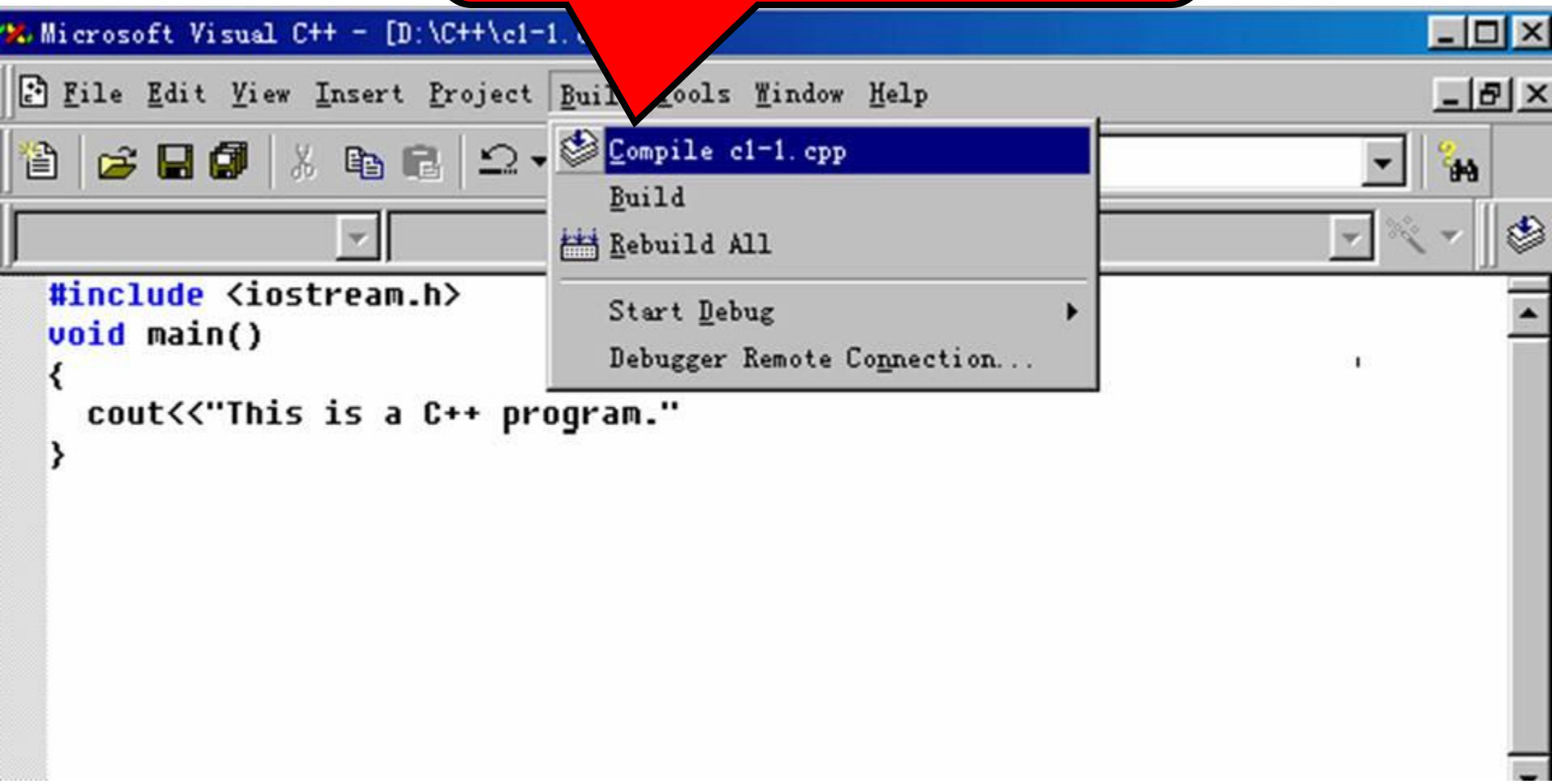

Microsoft Visual C++
File Edit View Insert Project Build Tools Window Help
单击“File”菜单中“New”命令
VC++编译系统界面
Ready

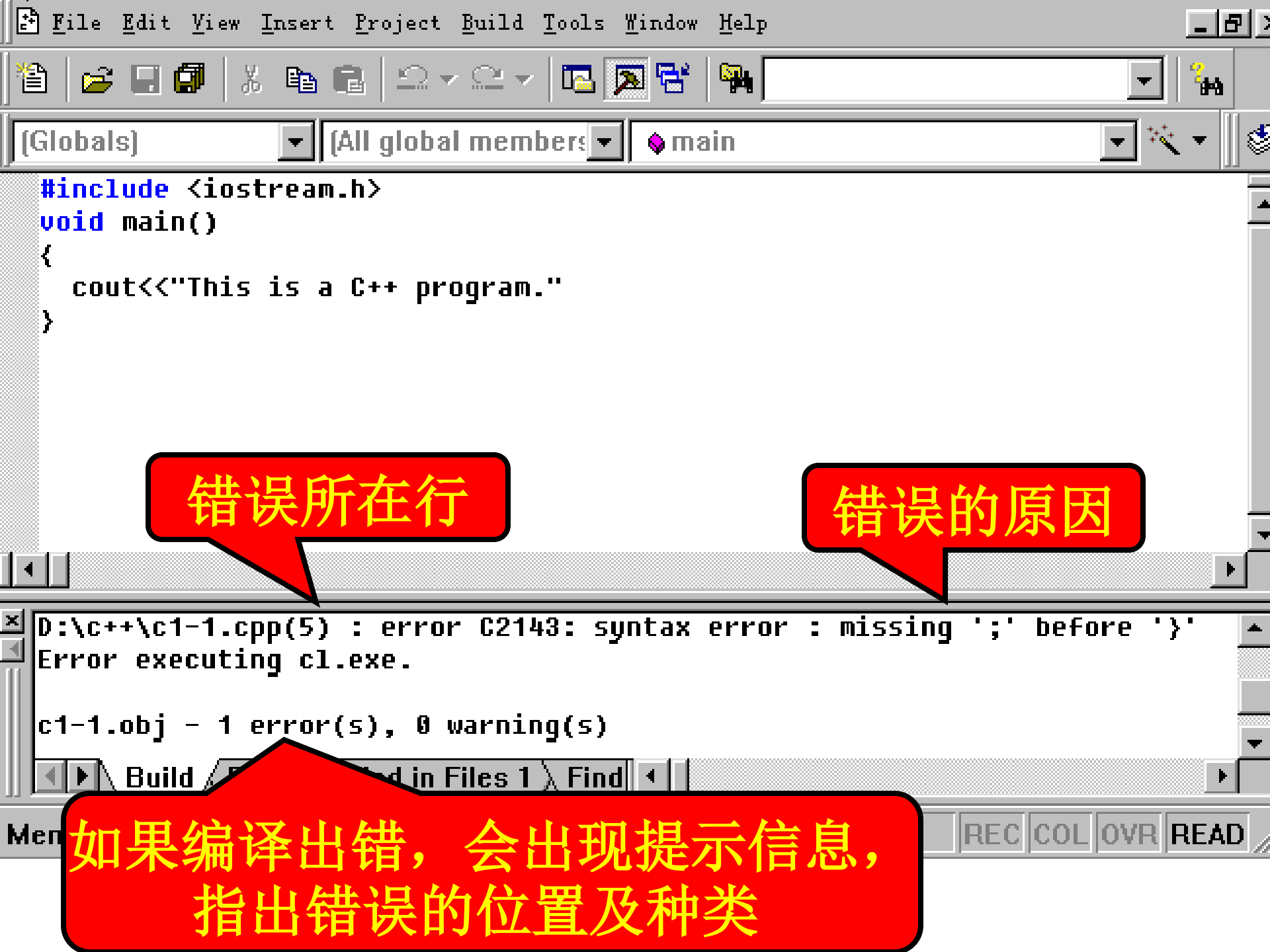

选择“Files”选项卡
New
Files
Projects
Workspaces
Other Documents
Active Server Page
Binary File
Bitmap File
C/C++ Header File
C++ Source File
Cursor File
HTML Page
Icon File
Macro File
Resource Script
选择C++源文件命令
Add to project:
输入文件名
File
c1-1.cpp
Location:
D:\C++
Choose Directory
Directory name:
C:\Documents and Settings\zhu
确定
结束
网络...
c:\
Documents and Settings
zhu
「开始」菜单
Cookies
Favorites
My Documents
桌面
驱动器:
c:
a:
c:
d: ZHU1
e:
f:
g:
输入文件存放位置
单击选择驱动器
选择驱动器或目录
OK
Cancel


Microsoft Visual C++ - [D:\C++\c1-1.cpp *]
File Edit View Insert Project Build Tools Window Help
#include <iostream.h>
void main()
{
cout<<"This is a C++ program."
}
输入C++源代码
C++源文件编辑界面
Build Debug Find in Files 1 Find
Ready
Ln 1, Col 1
REC COL OVR READ

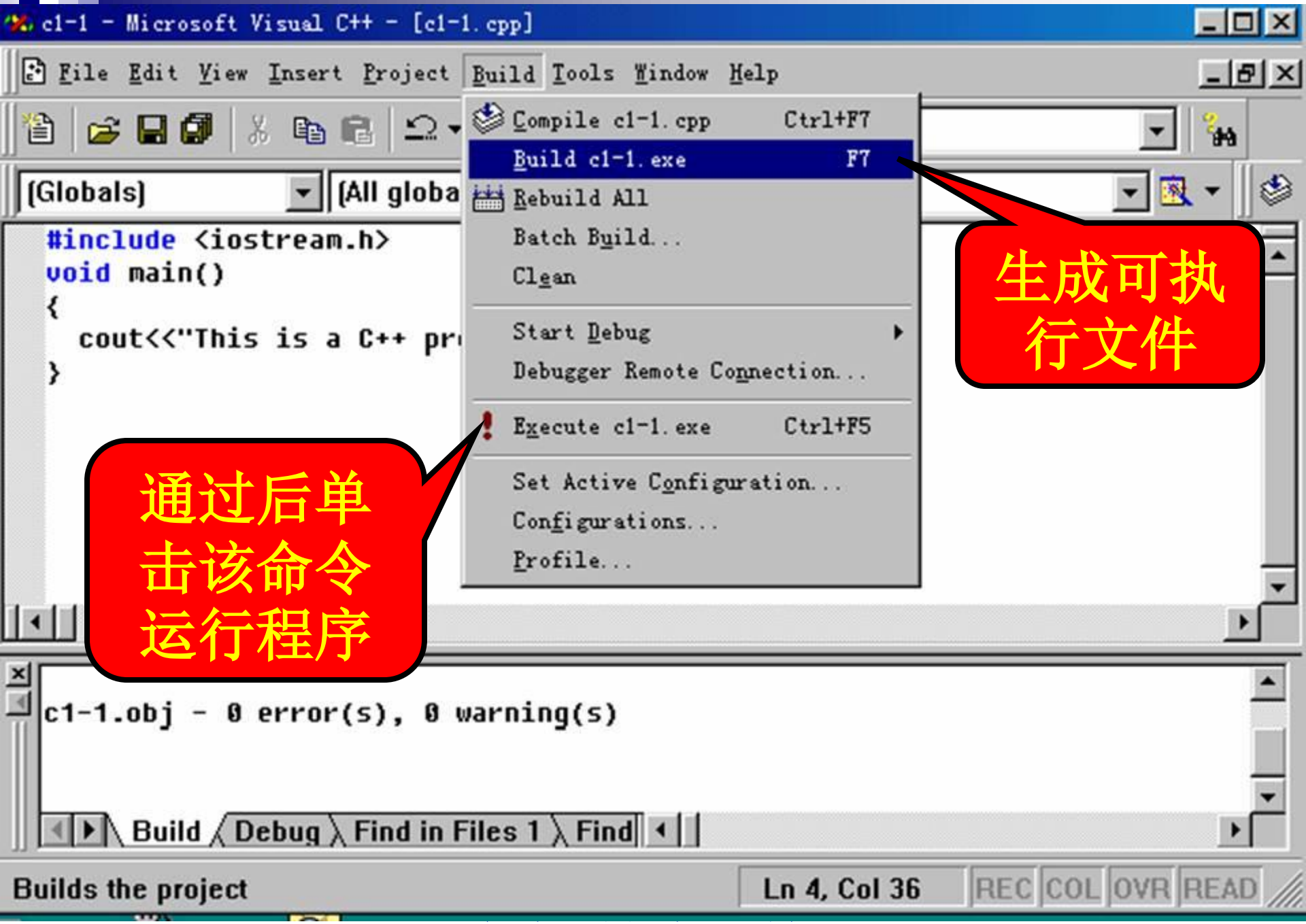

Microsoft Visual C++ - [D:\C++\c1-1.cpp *]
File Edit View Insert Project Build Tools Window Help
New... Ctrl+N
Open... Ctrl+O
Close
Open Workspace...
Save Workspace
Close Workspace
Save Ctrl+S
Save As...
Save All
Page Setup...
Print... Ctrl+P
Recent Files
Recent Workspaces
Exit
h>
C++ program."
可以将此源代码另起文件名存盘
Build Debug Find in Files 1 Find
Saves the document
Ln 2, Col 11
REC COL OVR READ

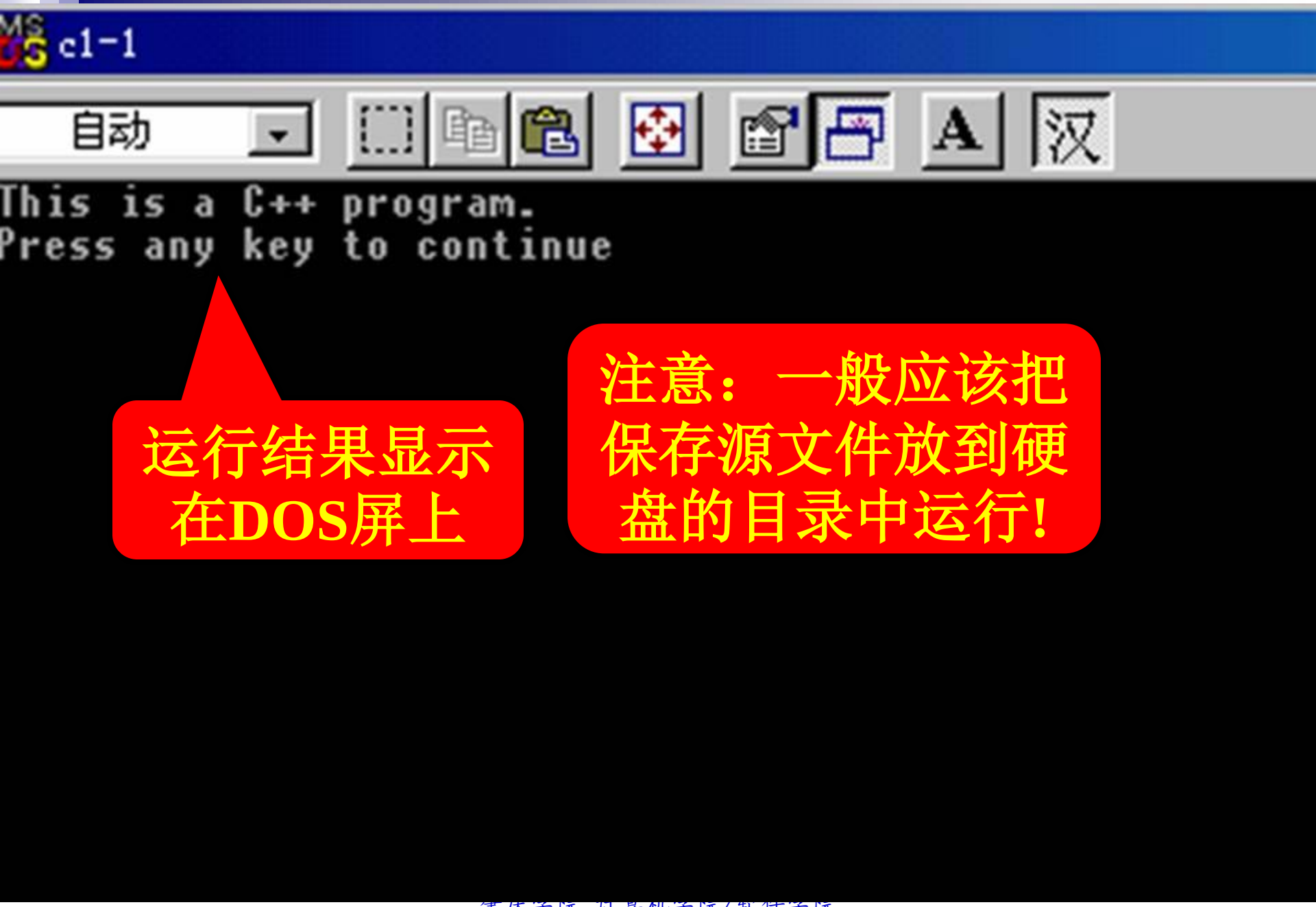
选择编译命令，将源文件.cpp生成.obj文件

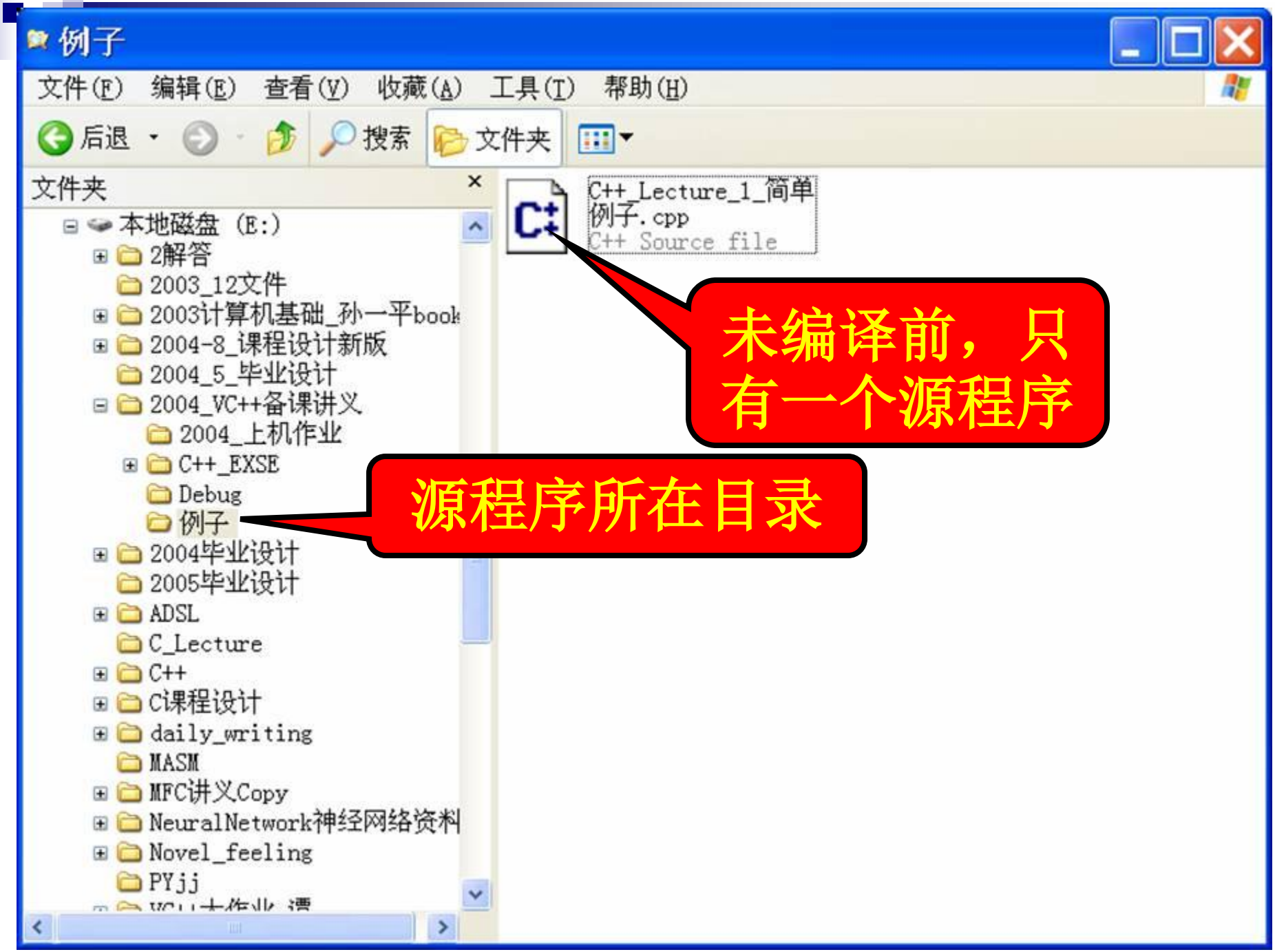


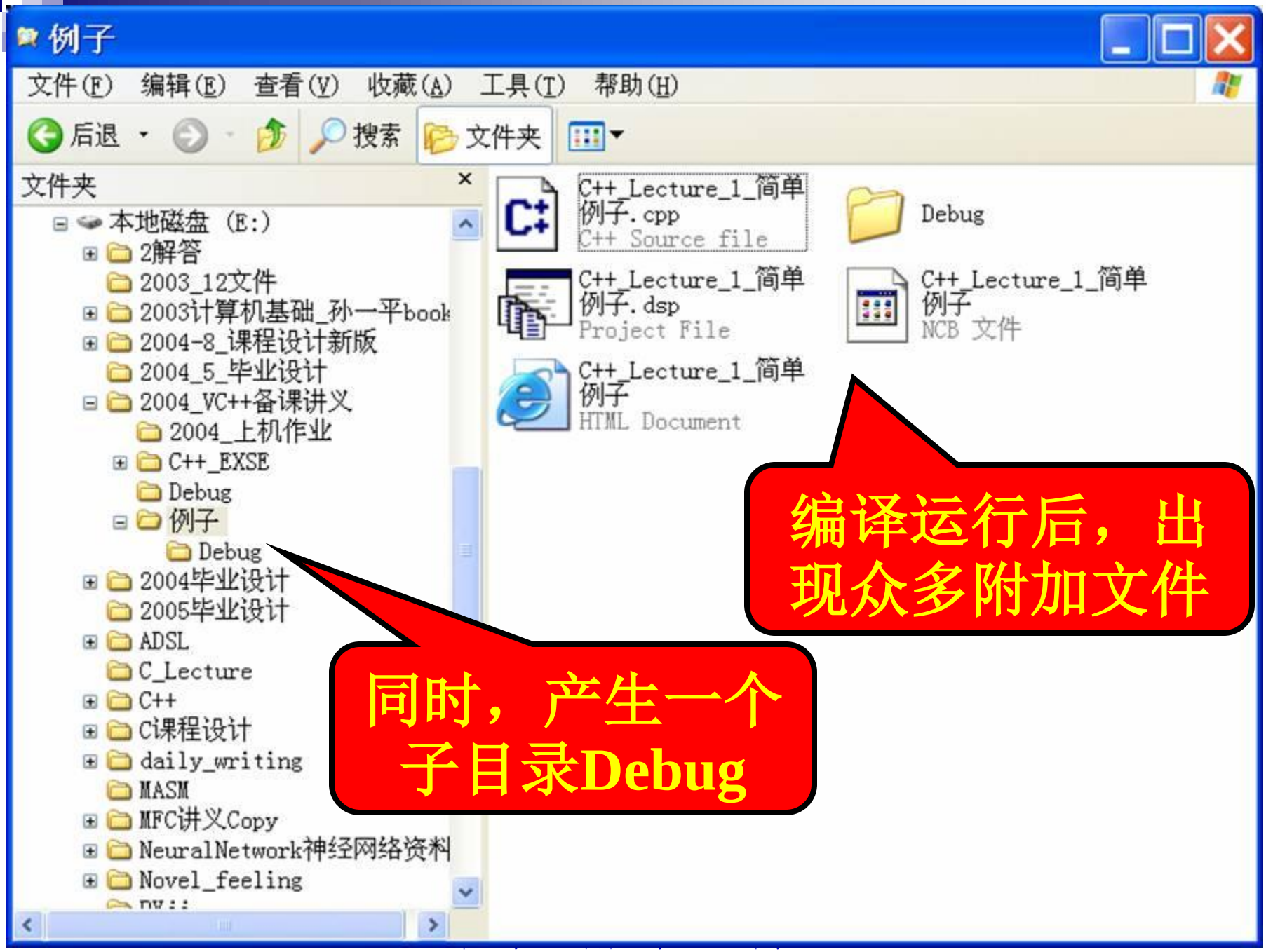

File Edit View Insert Project Build Tools Window Help
(Globals)
(All global members
main
#include <iostream.h>
void main()
{
cout<<"This is a C++ program."
}
错误所在行
错误的原因
D:\c++\c1-1.cpp(5) : error C2143: syntax error : missing ';' before '}'
Error executing cl.exe.
c1-1.obj - 1 error(s), 0 warning(s)
Build
Find
REC COL OVR READ
如果编译出错，会出现提示信息，
指出错误的位置及种类

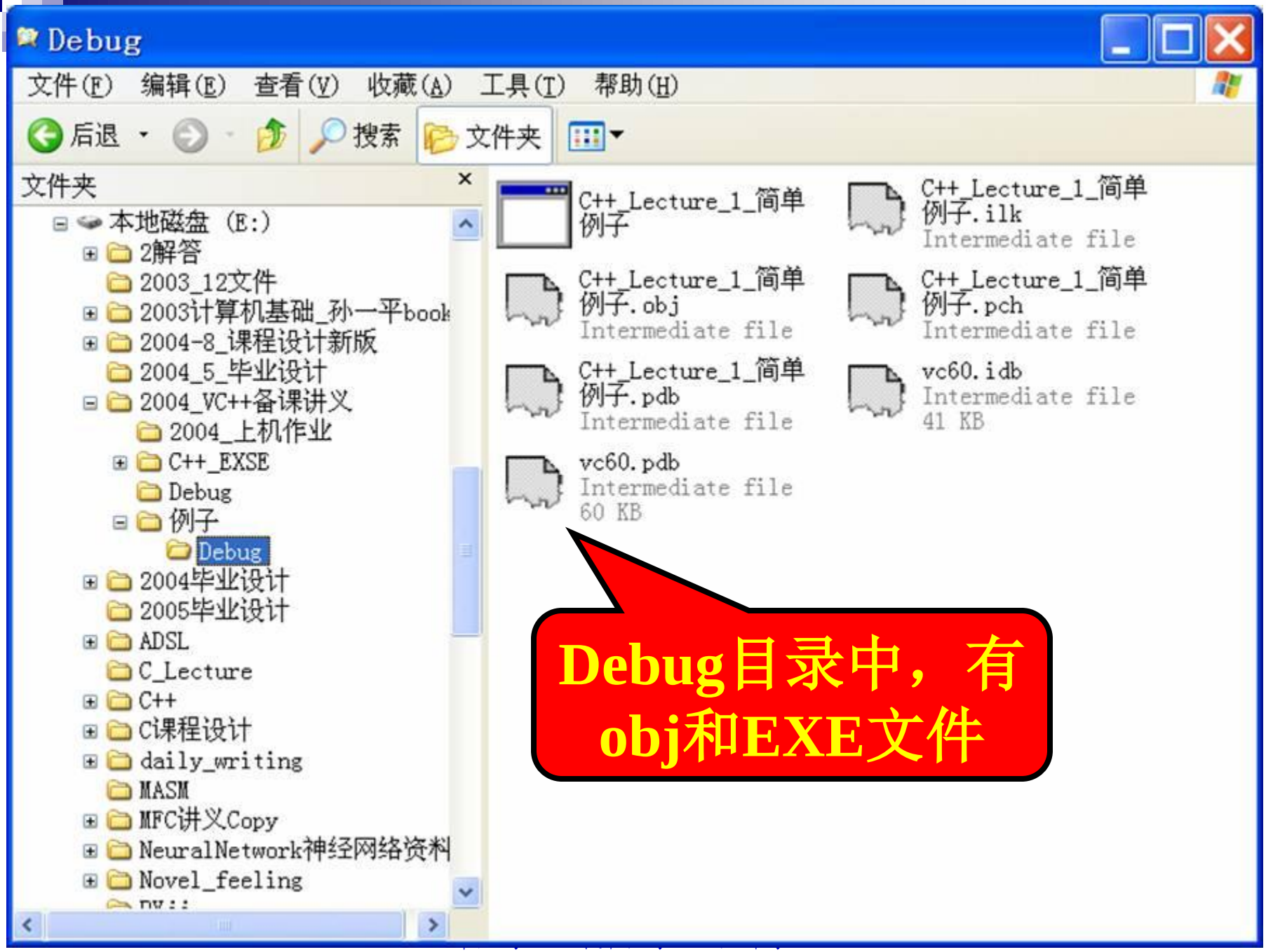

c1-1 - Microsoft Visual C++ - [c1-1.cpp]
File Edit View Insert Project Build Tools Window Help
(Globals)
(All global members
main
#include <iostream.h>
void main()
{
cout<<"This is a C++ program."
}
光标移到该行
双击错误
所在行
D:\c++\c1-1.cpp(5) : error C2143: syntax error : missing ';' before '}'
Error executing cl.exe.
c1-1.obj - 1 error(s), 0 warning(s)
Build
Debug
Find in Files 1
Find
Ready
Ln 5, Col 1
REC COL OVR READ


c1-1 - Microsoft Visual C++ - [c1-1.cpp]
File Edit View Insert Project Build Tools Window Help
Compile c1-1.cpp Ctrl+F7
Build c1-1.exe F7
Rebuild All
Batch Build...
Clean
Start Debug
Debugger Remote Connection...
Execute c1-1.exe Ctrl+F5
Set Active Configuration...
Configurations...
Profile...
(Globals)
#include <iostream.h>
void main()
{
cout<<"This is a C++ pr
}
生成可执行文件
通过后单击该命令运行程序
c1-1.obj - 0 error(s), 0 warning(s)
Build Debug Find in Files 1 Find
Builds the project
Ln 4, Col 36
REC COL OVR READ


c1-1
自动
This is a C++ program.
Press any key to continue
运行结果显示在DOS屏上
注意：一般应该把保存源文件放到硬盘的目录中运行!


例子
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 文件夹
文件夹
本地磁盘 (E:)
2解答
2003_12文件
2003计算机基础_孙一平book
2004-8_课程设计新版
2004_5_毕业设计
2004_VC++备课讲义
2004_上机作业
C++_EXSE
Debug
例子
2004毕业设计
2005毕业设计
ADSL
C_Lecture
C++
C课程设计
daily_writing
MASM
MFC讲义Copy
NeuralNetwork神经网络资料
Novel_feeling
PYjj
C++_Lecture_1_简单例子.cpp
C++ Source file
未编译前，只有一个源程序
源程序所在目录


例子
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 文件夹
文件夹
本地磁盘 (E:)
2解答
2003_12文件
2003计算机基础_孙一平book
2004-8_课程设计新版
2004_5_毕业设计
2004_VC++备课讲义
2004_上机作业
C++_EXSE
Debug
例子
Debug
2004毕业设计
2005毕业设计
ADSL
C_Lecture
C++
C课程设计
daily_writing
MASM
MFC讲义Copy
NeuralNetwork神经网络资料
Novel_feeling
C++_Lecture_1_简单例子.cpp
C++ Source file
Debug
C++_Lecture_1_简单例子.dsp
Project File
C++_Lecture_1_简单例子
NCB 文件
C++_Lecture_1_简单例子
HTML Document
编译运行后，出现众多附加文件
同时，产生一个子目录Debug


Debug
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 文件夹
文件夹
本地磁盘 (E:)
2解答
2003_12文件
2003计算机基础_孙一平book
2004-8_课程设计新版
2004_5_毕业设计
2004_VC++备课讲义
2004_上机作业
C++_EXSE
Debug
例子
Debug
2004毕业设计
2005毕业设计
ADSL
C_Lecture
C++
C课程设计
daily_writing
MASM
MFC讲义Copy
NeuralNetwork神经网络资料
Novel_feeling
C++_Lecture_1_简单例子
C++_Lecture_1_简单例子.ilk
Intermediate file
C++_Lecture_1_简单例子.obj
Intermediate file
C++_Lecture_1_简单例子.pch
Intermediate file
C++_Lecture_1_简单例子.pdb
Intermediate file
vc60.idb
Intermediate file
41 KB
vc60.pdb
Intermediate file
60 KB
Debug目录中，有obj和EXE文件

例2：

```
#include  <iostream.h>
void  main(void)
{
  cout << "i=";   //显示提示符
  int i;          //说明变量i
  cin  >>i;       //从键盘上输入变量i的值
  cout << "i的值为：" <<i<<'\n'; // 输出变量i的值
}
```

简单的输入输出

输入语句：cin

程序在执行期间，接收外部信息的操作称为程序的输入；而把程序向外部发送信息的操作称为程序的输出。在C++中没有专门的输入输出语句，所有输入输出是通过输入输出流来实现的。

要使用C++提供的输入输出时，必须在程序的开头增加一行：

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

即包含输入输出流的头文件“iostream.h”。

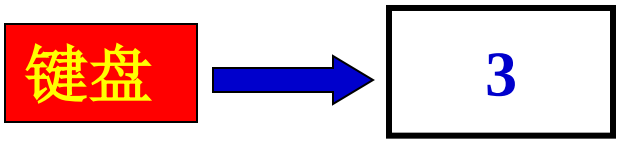
输入十进制整数

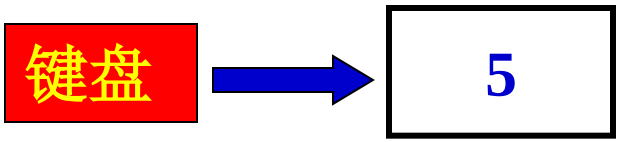
int a,b;

cin>>a>>b; //程序运行至此停下，等待从键盘输入变量值

键盘输入：3 5<CR>

或：3<CR> 5<CR> 均可。





输入语句自动过滤空白字符。

浮点型数据同整型数据一样。

```
float   c,d;

cin>>c>>d;

char   ch1,ch2;

cin>>ch1>>ch2;
```

若输入：ab<CR>　　则ch1为a, ch2为b。

若输入：a b<CR>　同样ch1为a, ch2为b。

字符型变量过滤空白字符。

```
float  a;
int  i1,i2;
char  ch1,ch2;
cin>>i1>>a>>i2>>ch1>>ch2;
```

输入：34 5.678 1a b<CR>

i1:34 a:5.578 i2:1 ch1:a ch2:b

在缺省的情况下，cin自动跳过输入的空格，

同样地，回车键也是作为输入字符之间的分隔符，也不能将输入的回车键字符赋给字符型变量。

若要把从键盘上输入的每一个字符，包括空格和回车键都作为一个输入字符赋给字符型变量时，必须使用函数cin.get()。其格式为：

cin.get(<字符型变量>);

cin.get()从输入行中取出一个字符，并将它赋给字符型变量。这个语句一次只能从输入行中提取一个字符。

```
char  c1;

cin.get(c1);
```

```
char  ch1,ch2,ch3;
cin.get(ch1);
cin.get(ch2);
cin.get(ch3);
```

则：ch1:A

ch2:空格

ch3:B

输入：A B<CR>

空格的ASCII码为32

ch2

0	0	1	0	0	0	0	0

输入十六进制或八进制数据

在缺省的情况下，系统约定输入的整型数是十进制数据。当要求按八进制或十六进制输入数据时，在cin中必须指明相应的数据类型：hex为十六进制；oct为八进制；dec为十进制。

例子见下页

```
int  i,j,k,l;
cin>>hex>>i;          //指明输入为十六进制数
cin>>oct>>j;          //指明输入为八进制数
cin>>k;           //输入仍为八进制数
cin>>dec>>l;          //指明输入为十进制数
```

当执行到语句cin时，若输入的数据为：

11 11 12 12<CR>

结果：i:17 j:9 k:10 l:12

使用非十进制数输入时，要注意以下几点：

1、八进制或十六进制数的输入，只能适用于整型变量，不适用于字符型变量，实型变量。

2、当在cin中指明使用的数制输入后，则所指明的数制一直有效，直到在接着的cin中指明输入时所使用的另一数制为止。如上例中，输入k的值时，仍为八进制。

3、输入数据的格式、个数和类型必须与cin中所列举的变量类型一一对应。一旦输入出错，不仅使当前的输入数据不正确，而且使得后面的提取数据也不正确。

```
int  a, b;
```

cin>>a,b; 错

cin>>a b;错

cin>>ab;错

cin>>a>>b; 正确

输出数据　　cout

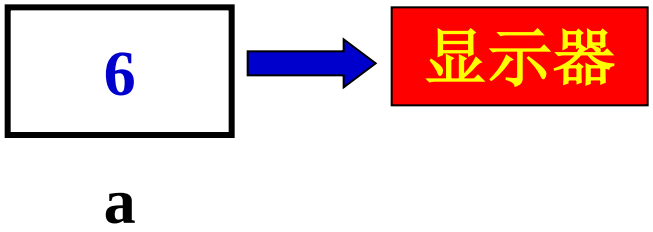
与输入cin对应的输出是cout输出流。

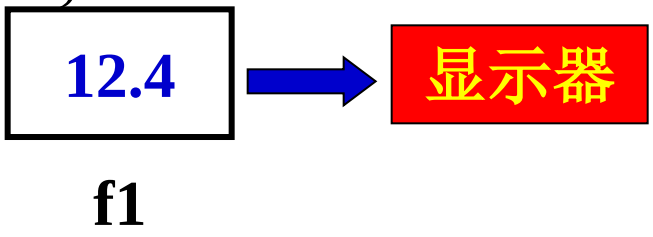
当要输出一个表达式的值时，可使用cout来实现，其一般格式为：

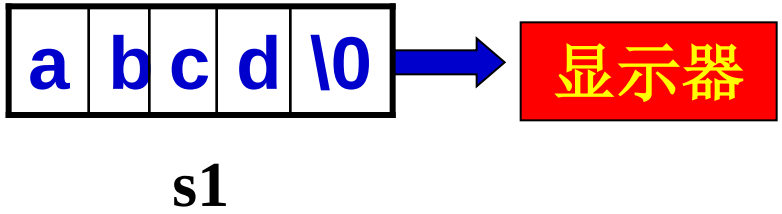
cout << <表达式> ；

其中运算符“<<”称为插入运算符，它将紧跟其后的表达式的值，输出到显示器当前光标的位置。

```
int a=6;

float  f1=12.4;

char  s1[ ]="abcd";

cout<<a<<'\t'<<f1<<'\t'<<s1<<endl;
```







'\t'为转义字符Tab

endl为回车或'\n'

cout将双引号中的字符串常量按其原样输出

```
char  ch1='a',ch2='b';
cout<<"c1="<<ch1<<'\t'<<"c2="<<ch2<<endl;
```

输出：c1=a　　c2=b

```
int  i1=4,i2=5;
float  a=3.5;
cout<<"a*i1="<<a*i1<<endl<<"a*i2="<<a*i2<<endl;
```

输出a*i1=14

a*i2=17.5

指定输出项占用的宽度：

在输出的数据项之间进行隔开的另一种办法是指定输出项的宽度。如上面的两个输出语句可改写为：

```
cout <<setw(6)<< i<<setw(10)<<j<<endl;

_ _ _ _ _ 4_ _ _ _ _ _ _ _ _ _12

cout << setw(5)<<m<<setw(10)<<j*k<<endl;

_ _ _ _ 7_ _ _ _ _ _ _ _ _ 24
```

其中setw(6)指明其后的输出项占用的字符宽度为6，即括号中的值指出紧跟其后的输出项占用的字符位置个数，并向右对齐。setw是“set width”的缩写。

使用setw()应注意以下三点：

1、在程序的开始位置必须包含头文件iomanip.h，即在程序的开头增加：

```
#include <iomanip.h>
```

2、括号中必须给出一个表达式（值为正整数），它指明紧跟其后输出项的宽度。

3、该设置仅对其后的一个输出项有效。一旦按指定的宽度输出其后的输出项后，又回到原来的缺省输出方式。

输出科学表示法的实数

对于实型数据可指定以科学表示法形式输出。例如，设有如下一个程序：

```
#include <iostream.h>
void main(void)
{    float x=3.14f,y=100;
     cout.setf(ios::scientific,ios::floatfield);
     //表明浮点数用科学表示法输出
     cout << x<<'\t';
     cout <<y<<endl;
}
```

执行该程序后的输出为：

3.140000e+000　　1.000000e+002

与cin中类同，当在cout中指明以一种进制输出整数时，对其后的输出均有效，直到指明又以另一种进制输出整型数据为止。对实数的输出，也是这样，一旦指明按科学表示法输出实数，则接着的输出均按科学表示法输出，直到指明以定点数输出为止。明确指定按定点数格式输出（缺省的输出方式）的语句为：

```
cout.setf(ios::fixed,ios::floatfield);
```

//按上页例子比较一下

作业：

每两周交一次，提交至ftp://172.21.85.11,用户名:16kjhomework,密码:11111111，提交要求：请登录FTP查看。

- P35
- 第11，13，14题