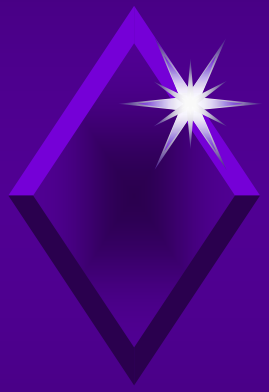
C++高级语言程序设计



课程介绍

- 教学形式
  - 课堂讲授+上机实验
  - 讲课：
    - 每周2学时，不得缺席。
  - 实验课：
    - 每2周2学时，不得缺席，有助教辅导答疑。
- 考核要求
  - 必修考试课（闭卷）
  - 成绩组成：
    - 平时成绩 20%
      - 作业(按时提交,不提交无成绩)10%
      - 实验（当堂检查评分,不提交不给成绩）10%
    - 笔试 ： 期中10%， 期末70%

学习建议

- 不同于传统的学习方法（数学、物理）
- 实验性强

  在阅读教材时，应该边阅读、边实践。

  有条件的，应该坐在计算机前，边阅读边亲自编写每一个例题程序，如果对于某些概念、语法存有疑问，应该立即编写程序予以验证。在完全理解了主教材内容以后，再开始做实验和习题。
- 课堂讲过的内容要掌握
- 知识拓展，阅读参考书

课程介绍

教材

《C++程序设计》
清华大学出版社，谭浩强 编著

参考书

《C++程序设计题解与上机指导》
清华大学出版社，谭浩强 编著

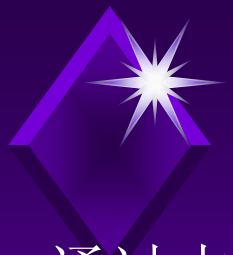
- C++Primer中文版
  Stanley B.Lippman Josee Lajoie BARbara E.Moo著 人民邮电出版社出版
- 《Visual C++ 6.0 使用与开发》，木林森，高峰霞等编著，清华大学出版社

课程介绍

- 课程安排　　48学时 (其中上机实验16学时)
- 上课时间

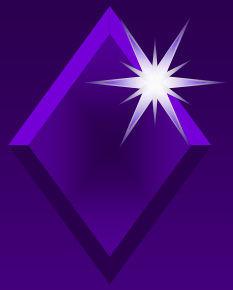
  周二7-8节 3:30-5:20上课

  周四5-6节 1：30-3：20

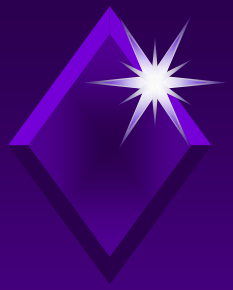


课程教学目的

- 通过本课程，使学生具备基本的C++程序设计能力。
- 包括：
  - 掌握C++语言的基本语法、基本概念、和基本原理；
  - 理解和掌握应用C++语言进行面向对象程序设计的基本思想和方法；
  - 掌握编程和调试程序的一般方法；
  - 提高抽象思维能力，初步掌握应用面向对象的思想和方法求解实际问题的方法。



第一章 绪论



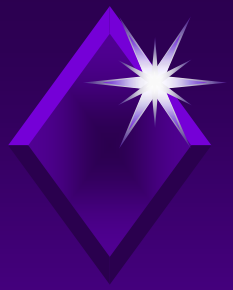
主要内容

- 计算机程序设计语言的发展
- 计算机程序设计方法的发展

计算机语言的发展

- 计算机系统
  - 硬件
    - 运算器
    - 控制器
    - 存储器
    - 输入设备
    - 输出设备
  - 软件
    - 系统软件
    - 应用软件

软件=程序+文档



计算机程序

计算机语言的发展

- 计算机的工作是用程序来控制的
- 程序是指令的集合。
- 指令是计算机可以识别的命令。

计算机语言的发展

- *计算机语言*
  - 低级语言
    - 机器语言
    - 汇编语言
  - 高级语言
    - 面向过程：C语言
    - 面向对象：C++，JAVA
  - 分类：编译型，解释型

机器语言与汇编语言

- 由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言。

  计算机发展的初期，软件工程师们只能用机器语言来编写程序。这一阶段，在人类的自然语言和计算机编程语言之间存在着巨大的鸿沟。

- 汇编语言将机器指令映射为一些可以被人读懂的助记符，如ADD、SUB等。

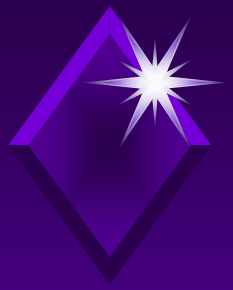
  此时编程语言与人类自然语言间的鸿沟略有缩小，但仍与人类的思维相差甚远。因为它的抽象层次太低，程序员需要考虑大量的机器细节。

高级语言

高级语言屏蔽了机器的细节，提高了语言的抽象层次，程序中可以采用具有一定含义的数据命名和容易理解的执行语句。这使得在书写程序时可以联系到程序所描述的具体事物。缩小人类与机器的鸿沟

第一种高级语言的诞生于1954年，用于科学计算的FORTRAN语言，后又出现了多种，如BASIC、ALGOL、PASCAL、COBOL、ADA和C语言

C语言是1972年由美国贝尔实验室的D.M.Ritchie研制成功的。特点是面向过程的结构化程序设计语言



面向对象的语言

计算机语言的发展

- 更直接地描述客观世界中存在的事物 (对象)以及它们之间的关系。
- 特点：
  - 是高级语言。
  - 将客观事物看作具有属性和行为的对象。
  - 通过抽象找出同一类对象的共同属性和行为，形成类。
  - 通过类的继承与多态实现代码重用

15

面向对象的语言

- 优点：

使程序能够比较直接地反映问题域的本来面目，软件开发人员能够利用人类认识事物所采用的一般思维方法来进行软件开发。

面向过程的结构化程序设计方法

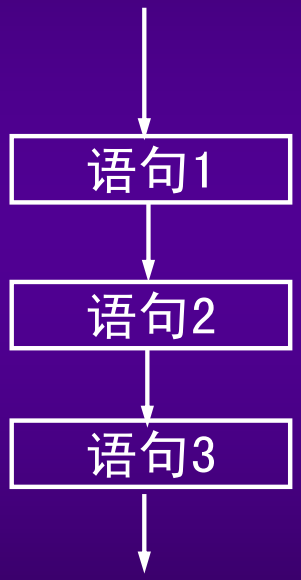
程序设计方法的发展

结构化方法出现在70年代中期，我们可以这样理解它：

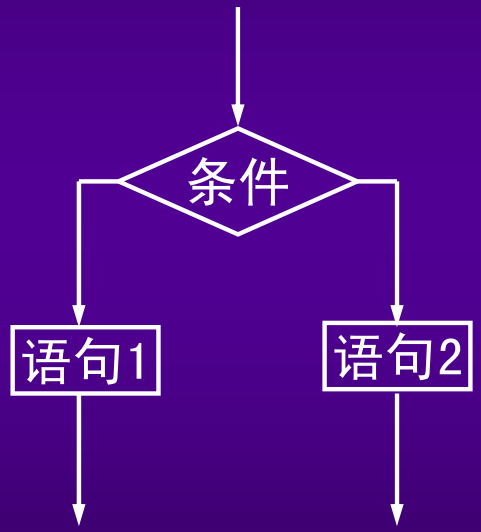
- 结构化程序设计方法是从程序要实现的功能的角度出发的。一般按照自顶向下、逐步求精的方式，将程序要完成的功能逐级划分成许多小的功能模块，象搭积木一样搭起来。
- 这些小的功能模块最终都可以转化成三种基本控制结构的组合。

面向过程的结构化程序设计方法

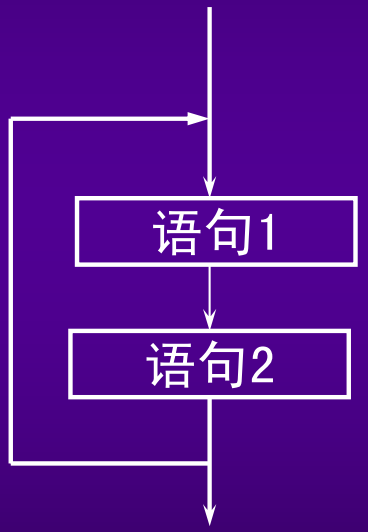
- 顺序结构

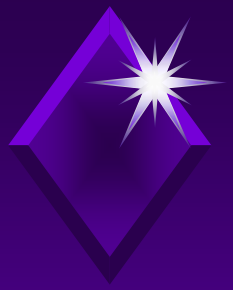


- 分支结构



- 循环结构





一个简单的例子

例：

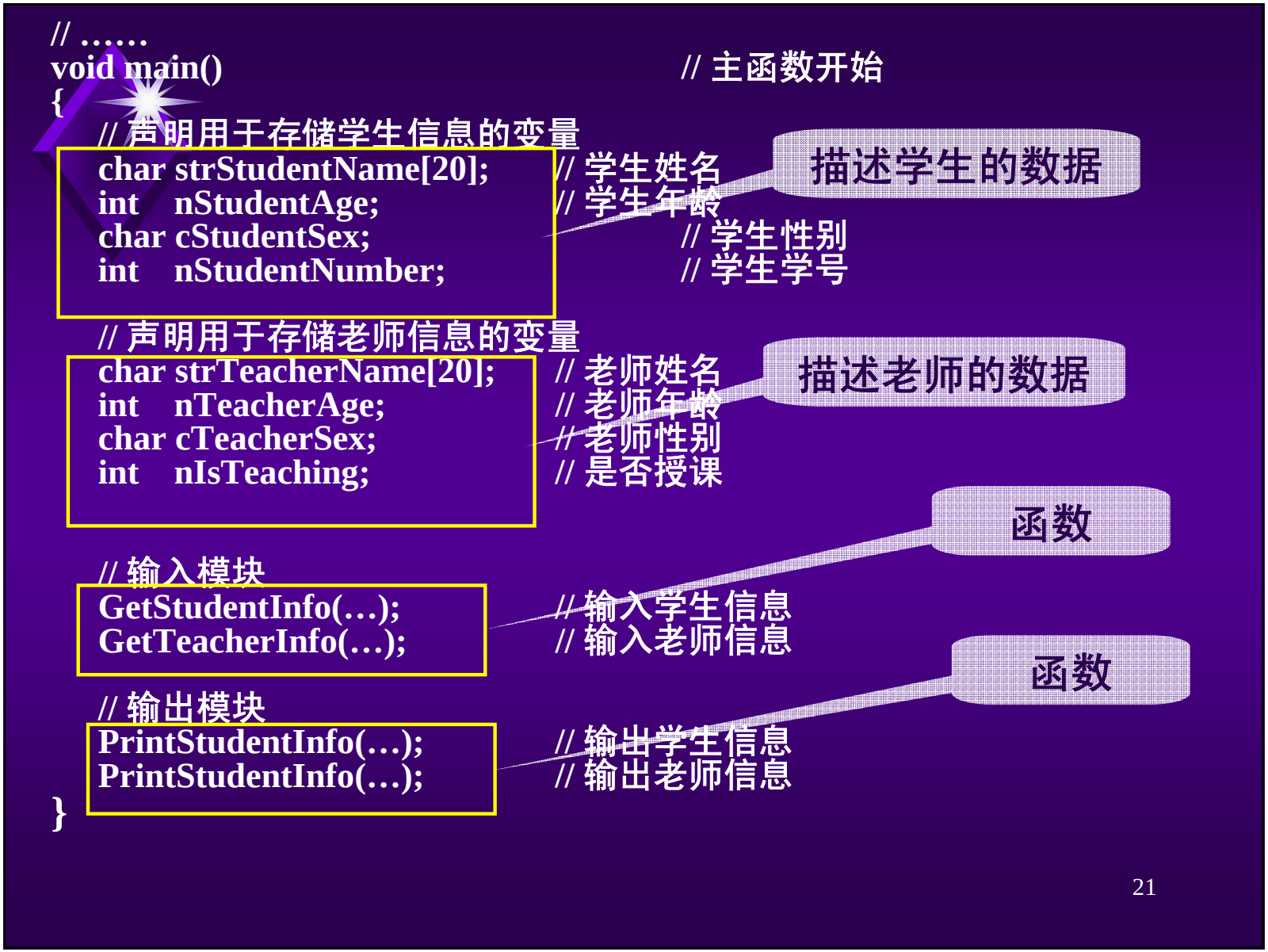
从键盘输入一个学生的信息（包括姓名、年龄、性别、学号等）和一个老师的信息（包括姓名、年龄、性别、是否授课等），然后将信息输出到屏幕。

分析：

根据需求（题目要求），我们可以把问题划分为两个功能模块，一个是输入模块，一个是输出模块，

再具体考虑每个模块如何实现（逐步求精）。

我们用C语言来写，参看下面的代码：

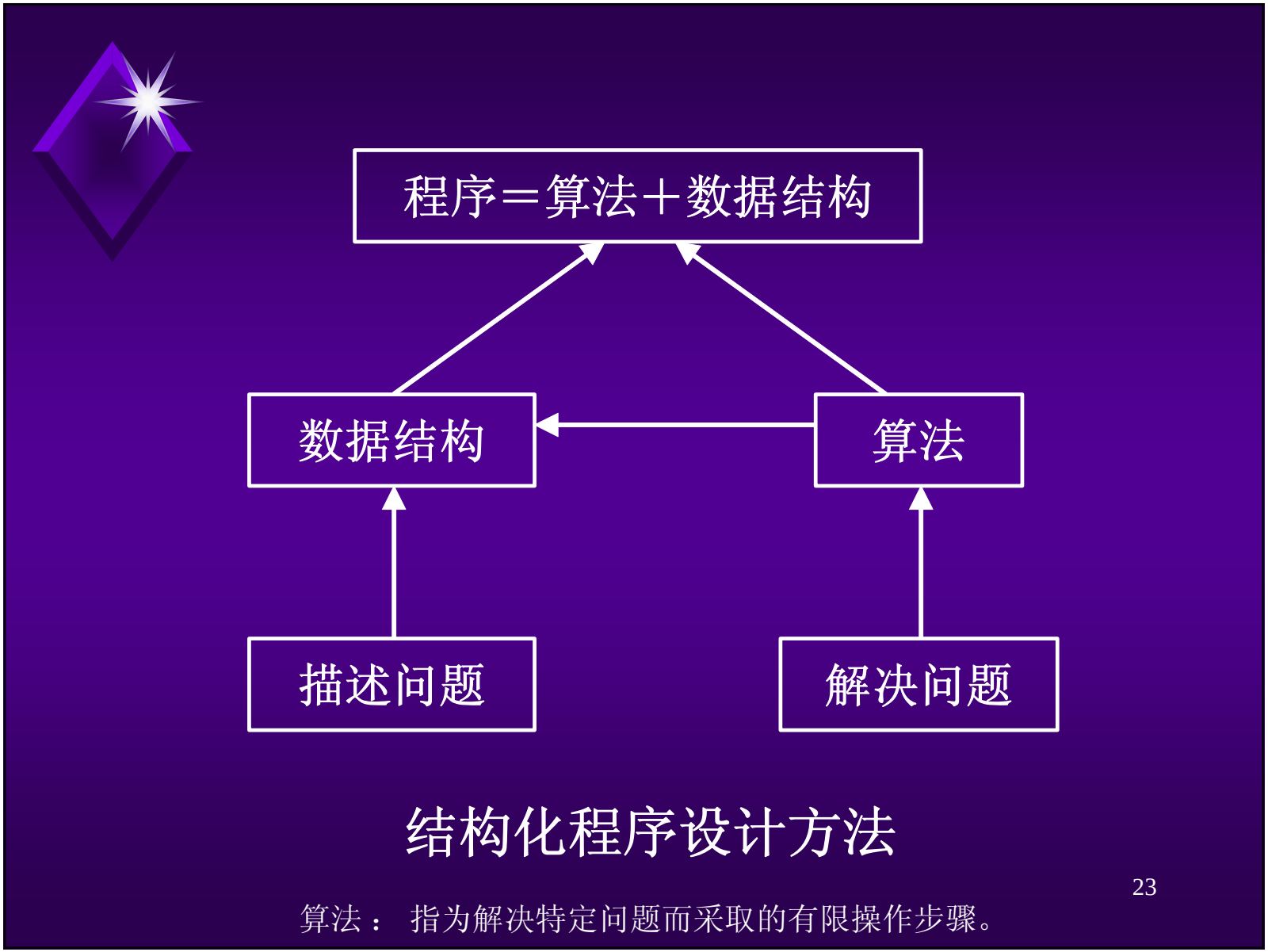

// ……
void main()
{
// 声明用于存储学生信息的变量
char strStudentName[20];
int nStudentAge;
char cStudentSex;
int nStudentNumber;
// 主函数开始
// 学生姓名
// 学生年龄
// 学生性别
// 学生学号
描述学生的数据
// 声明用于存储老师信息的变量
char strTeacherName[20];
int nTeacherAge;
char cTeacherSex;
int nIsTeaching;
// 老师姓名
// 老师年龄
// 老师性别
// 是否授课
描述老师的数据
// 输入模块
GetStudentInfo(…);
GetTeacherInfo(…);
// 输入学生信息
// 输入老师信息
函数
// 输出模块
PrintStudentInfo(…);
PrintStudentInfo(…);
// 输出学生信息
// 输出老师信息
函数
}

21

上面的例子中，我们可以进一步将属于学生和老师的变量放入结构中。这样可以在一定程度上完成对数据的封装。但在结构化程序设计中，数据与对其进行操作的函数仍是分离的。

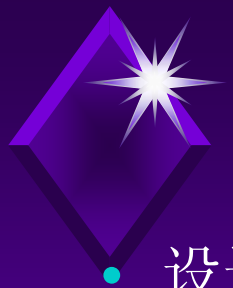
```
// 声明学生结构Student
struct Student
{
    char strStudentName[20];      // 学生姓名
    int    nStudentAge;           // 学生年龄
    char cStudentSex;                    // 学生性别
    int    nStudentNumber;               // 学生学号
};
// 声明老师结构Teacher
struct Teacher
{
    char strTeacherName[20];      // 老师姓名
    int    nTeacherAge;           // 老师年龄
    char cTeacherSex;             // 老师性别
    int    nIsTeaching;           // 是否教书
};
```


22



结构化程序设计方法

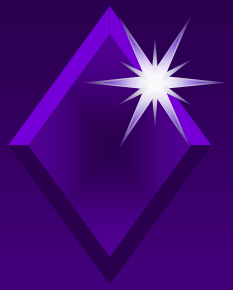
算法： 指为解决特定问题而采取的有限操作步骤。



面向过程的结构化程序设计方法

- 设计步骤：
  - 提出问题—如何使用某种语言设计实现解决问题的方案？
  - 将它分解为多个便于处理的部分--模块
  - 设计许多数据结构 在其中保存数据
  - 实现许多函数（也成为过程）以对这些数据进行操作
  - 函数修改数据结构，将它们保存到文件中并打印数据，
  - 我们对他们的 所有了解都转换成了一组函数，工作的 重点就是函数，因为没有它们就 无法完成任何有用的操作
  - 总之，重点在于过程，以函数形式思考问题，也称问题的功能分解

24



面向过程的结构化程序设计方法

- 缺点：
  - 可重用性差、数据安全性差、难以开发大型软件和图形界面的应用软件
  - 把数据和处理数据的过程分离为相互独立的实体。
  - 当数据结构改变时，所有相关的处理过程都要进行相应的修改。

问题：

函数用于完成一定的功能，它们都是针对特定的数据进行操作的。那么我们能不能以特定的数据为中心，将数据与对其进行操作的函数封装起来呢？

*面向*对象*的程序设计方法*

面向对象程序设计出现在80年代中后期

- 面向对象程序设计是建立在结构化程序设计基础上的，但它不再是从功能入手，不是以函数过程和数据结构为中心,而是从对象入手,是以对象代表问题的中心环节，
- 将数据及对数据的操作方法封装在一起，作为一个相互依存、不可分离的整体——对象。

- 对同类型对象抽象出其共性，形成类。
- 类通过一个简单的外部接口，与外界发生关系。
- 对象与对象之间通过消息进行通信。

例：C++中类的声明——Student类

```
class Student                          // Student类的声明
{
public:                                // 公有成员
    Student();                         // 构造函数
    ~Student();                        // 析构函数

    char*  GetName();                  // 查询姓名
    int    GetAge();                   // 查询年龄
    char   GetSex();                   // 查询性别
    int    GetNumber();                // 查询学号

    bool   SetName(char* n);           // 设置姓名
    bool   SetAge(int age);            // 设置年龄
    bool   SetSex(char* s);            // 设置性别
    bool   SetNumber(int num);         // 设置学号
protected:                             // 保护成员
    char   m_strName[20];              // 姓名，字符串数组
    int    m_nAge;                     // 年龄，整型
    char   m_cSex;                     // 性别，字符型
    int    m_nNumber;                  // 学号，整型
};
```

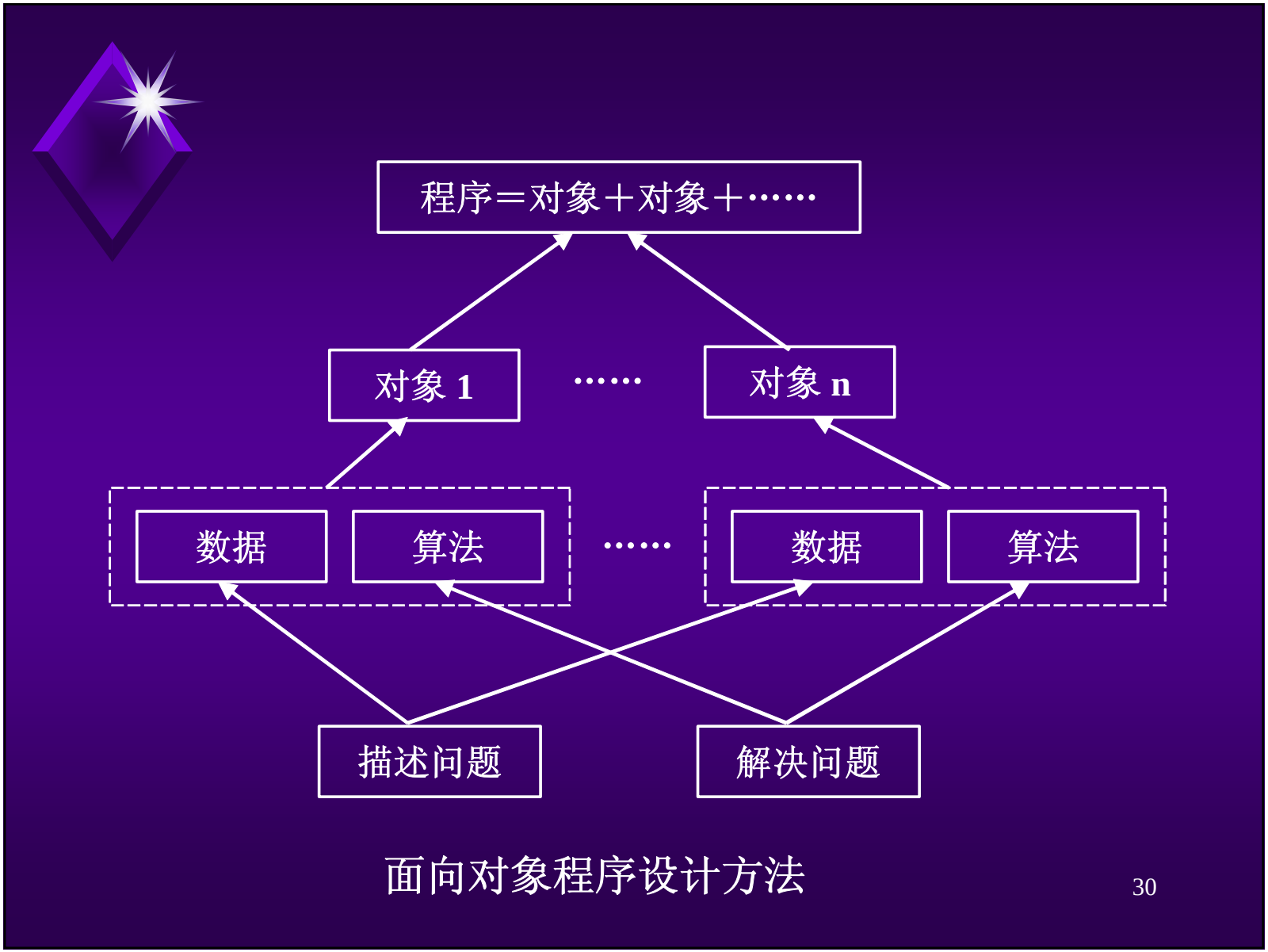
成员函数

成员函数

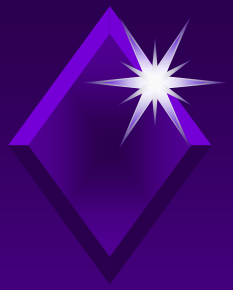
成员变量

例：C++中类使用

```
……
Student A;                    // 声明Student的对象A
A.SetName(“张三”);            // 设置A的名字
A.SetAge(20);                 // 设置A的年龄
……
```



面向对象程序设计方法



总的来说:

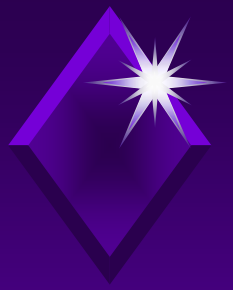
- 结构化程序设计方法它在解决问题时是以功能为中心的，一定的功能模块虽然也作用于特定的数据，但它们并没有被封装在一起。
- 面向对象程序设计方法则是以对象为中心来解决问题的。属于同种对象的属性（数据）和服务（功能）被抽象出来封装到一起。

31

面向对象的程序设计方法

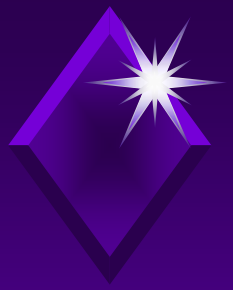
程序设计方法的发展

- 优点：
  - 符合人们对客观世界的认识规律
  - 对需求变化具有很强的适应性
  - 支持软件复用
  - 可维护性好



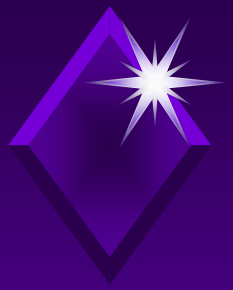
面向对象的基本概念

- 对象
- 类
- 封装
- 继承
- 消息
- 多态性



面向对象的基本概念

- “面向对象”（Object-Oriented，OO）？
- 软件工程学家PeterCoad和EdwardYourdon在1991年提出了如何识别面向对象方法的标准：
- **面向对象=对象(objects)**
  +类(classes)
  +继承(inheritance)
  +消息通信
  (communication with messages)
- 如果一个计算机软件系统采用这些概念来建立模型并实现，它就是面向对象的。



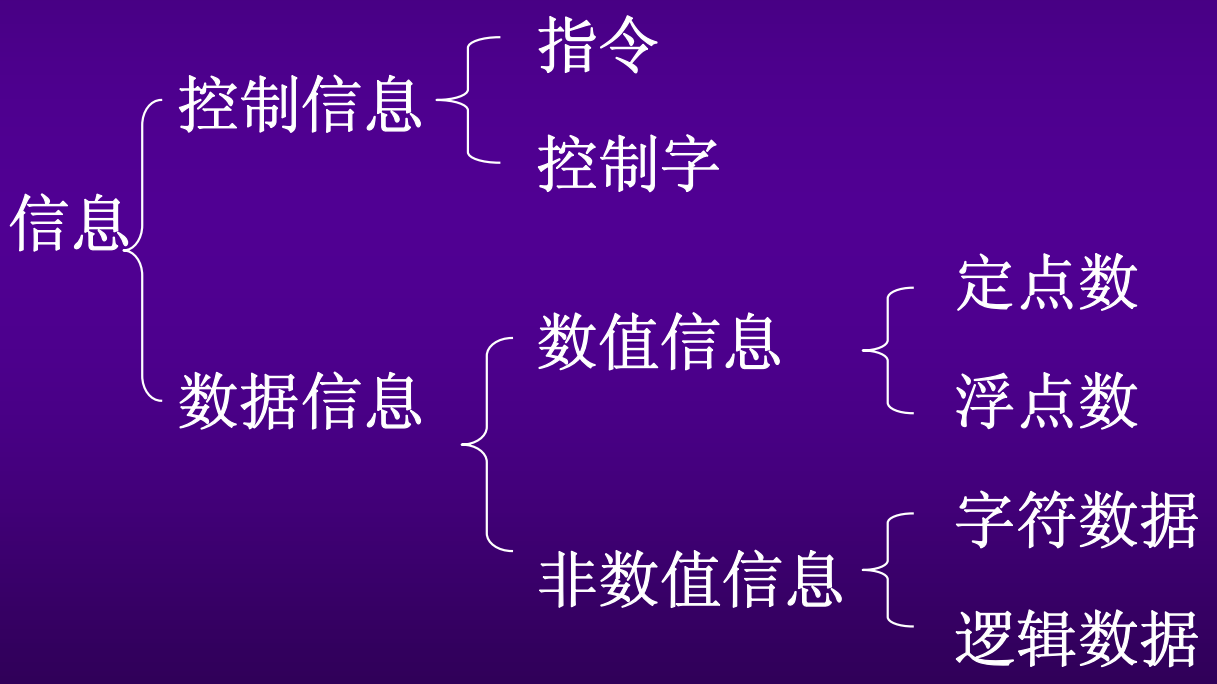
信息的表示和存储

- 信息的分类
- 计算机的数字系统
- 程序设计中常用的数制
- 不同进位计数制间的转换
- 信息的存储单位
- 二进制数的编码表示
- 小数的表示方法
- 非数值信息的表示

35

信息的表示与存储

信息的分类



信息的表示与存储

程序设计中常用的数制

进制	基 数	进位原则	基本符号
二进制	2	逢 2 进 1	0, 1
八进制	8	逢 8 进 1	0, 1, 2, 3, 4, 5, 6, 7
十进制	10	逢 10 进 1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
十六进制	16	逢 16 进 1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

不同进位记数制间的转换
——二进制→十进制

各位数字与它的权相乘，其积相加。

例如：

$(11111111.11)_2=1\times2^7+1\times2^6+1\times2^5+1\times2^4+1\times2^3+1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}=(255.75)_{10}$

$(3506.2)_8=3\times8^3+5\times8^2+0\times8^1+6\times8^0+2\times8^{-1}=(1862.25)_{10}$

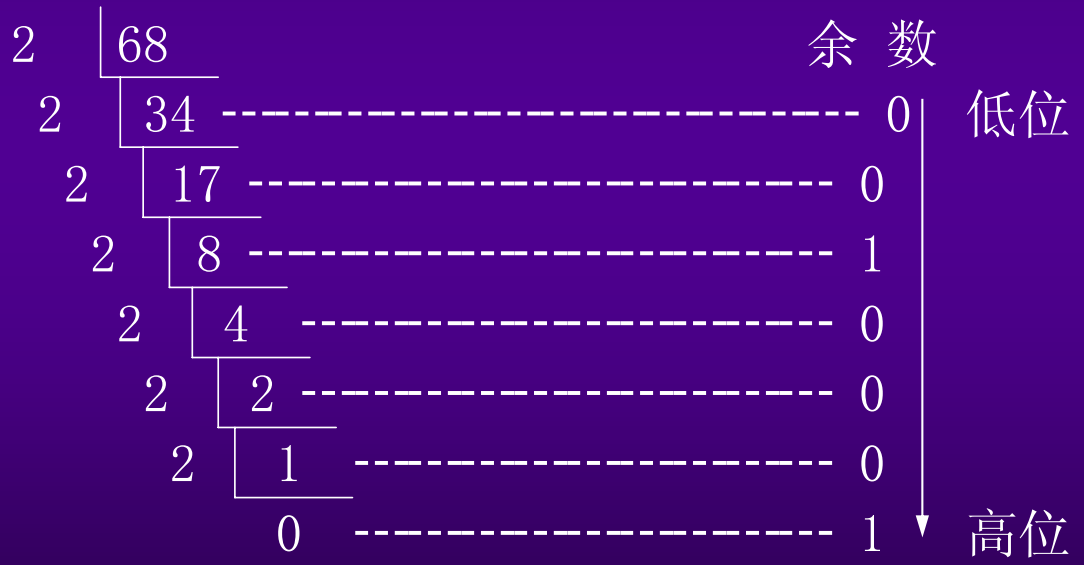
$(0.2A)_{16}=2\times16^{-1}+10\times16^{-2}=(0.1640625)_{10}$

38

不同进位记数制间的转换
——十进制→二进制

十进制整数转换成二进制的整数

“除二取余”法，例如：



所以 $68_{10}=1000100_2$

不同进位记数制间的转换
——十进制→二进制

信息的表示与存储

十进制小数转换成二进制小数

“乘 二取整”法，例如：

高位

$0.3125 \times 2 = \boxed{0}.625$

$0.625 \times 2 = \boxed{1}.25$

$0.25 \times 2 = \boxed{0}.5$

$0.5 \times 2 = \boxed{1}.0$

所以 $0.3125_{10} = 0.0101_2$

不同进位记数制间的转换
——二、八、十六进制的相互转换

- 每位八进制数相当于三位二进制数
- 每位十六进制数相当于四位二进制数

$(1011010.10)_2=(\underline{001}\ \underline{011}\ \underline{010}\ .\underline{100})_2$
$=(132.4)_8$

$(1011010.10)_2=(\underline{0101}\ \underline{1010}\ .\underline{1000})_2$
$=(5A.8)_{16}$

$(F7)_{16}=(\underline{1111}\ \underline{0111})_2=(11110111)_2$

信息的存储单位

- 位(bit，b)：度量数据的最小单位，表示一位二进制信息。
- 字节(byte，B)：由八位二进制数字组成(1 byte = 8 bit)。

千字节	1 KB = 1024 B
兆字节	1 MB = 1024 K
吉字节	1 GB = 1024 M

42

二进制数的编码表示:原码

- "符号——绝对值表示"的编码

  例如:

  X=+0101011　　$[X]_{原}$= 0 0101011

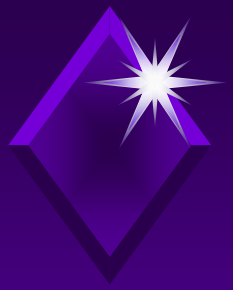
  X=−0101011　　$[X]_{原}$= 1 0101011

  符号位

- 缺点:
  - 零的表示不惟一:

    $[+0]_{原}$ =000...0　$[-0]_{原}$ =100...0
  - 进行四则运算时，符号位须单独处理，且运算规则复杂。

43



二进制数的编码表示:反码

信息的表示与存储

- 正数的反码与原码表示相同。
- 负数的反码与原码有如下关系：

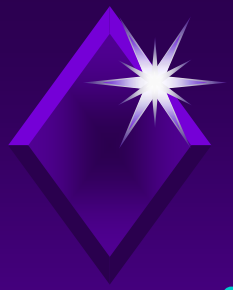
  符号位相同(仍用1表示)，其余各位取反(0变1，1变0)。例如：

  $X=-1100110\ [X]_{原}=11100110\ [X]_{反}=10011001$

  $X=+0000000\ [X]_{原}=00000000\ [X]_{反}=00000000$
- 反码中零的表示也不惟一

  $X=-0000000\ [X]_{原}=10000000\ [X]_{反}=11111111$
- 反码只是求补码的中间码

44



二进制数的编码表示:补码

信息的表示与存储

- 计算机中的补码表示法
  - 负数的补码由该数反码的末位加 1 求得
  - 对补码再求补即得到原码
- 补码运算规则
  - 符号位可作为数值参加运算
  - 减法运算可转换为加法运算：
    加上一个负数等于加上该数的补码
  - 补码运算的结果仍为补码
  - 运算结果溢出：
    负数之和得正数，或正数之和得负数

45

信息的表示与存储

实数的表示方法

- 计算机中通常采用浮点方式表示实数
一个数 N 用浮点形式表示可以写成：

$$N=M\times 2^{E}$$

  - E表示2的幂，称为数N的阶码。阶码确定了数N的小数点的位置，其位数影响该浮点数所表示的范围。
  - M表示数N的全部有效数字，称为数N的尾数。其位数影响数据的精度。

46

信息的表示与存储

非数值信息的表示

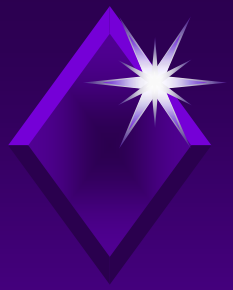
- 西文字符：
  - ASCII码：用7位二进制数表示一个字符，最多可以表示2^7=128个字符
  - EBCDIC码：用8位二进制数表示一个字符，最多可以表示2^8=256个字符
- 汉字：
  - 应用较为广泛的是"国家标准信息交换用汉字编码"(GB2312-80标准)，简称国标码。是二字节码，用二个七位二进制数编码表示一个汉字。

47

程序的开发过程

基本术语

- 源程序：
  - 用源语言写的，有待翻译的程序
- 目标程序：
  - 也称为“结果程序”，是源程序通过翻译程序加工以后所生成的二进制程序。
- 翻译程序：
  - 是指一个把源程序翻译成等价的目标程序的程序。



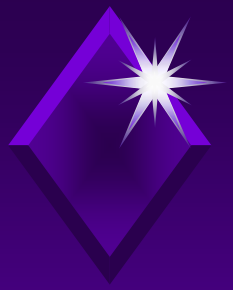
小结与复习建议

- 简要介绍了如下内容
  - 计算机程序设计语言的发展 、面向对象的方法、信息的表示与存储 、程序的开发过程
- 达到的目标
  - 初步了解面向对象的程序设计语言之由来，初步了解面向对象的程序设计思想之基本特点，概要性地了解面向对象的软件开发方法，为后续章节的学习奠定基础。
- 实验任务
  - 实验一

49

C++高级语言程序设计

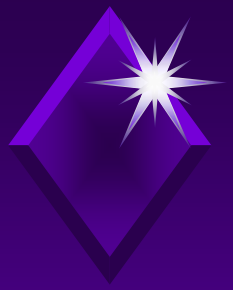
第二章 C++简单程序设计



本章主要内容

- C++语言概述
- 最简单的C++程序
- C++程序的构成和书写形式
- C++程序的编写和实现
- 基本数据类型和表达式

51

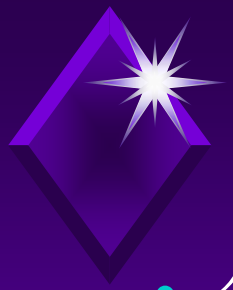


C++语言概述

C++语言的产生

- C++是从C语言发展演变而来的，首先是一个更好的C
- 引入了类的机制，最初的C++被称为“带类的C”
- 1983年正式取名为C++
- 从1989年开始C++语言的标准化工作
- 于1994年制定了ANSI C++标准草案
- 于1998年11月被国际标准化组织（ISO）批准为国际标准，成为目前的C++

52



C++语言概述

C++的特点

- 全面兼容C
  - 它保持了C的简洁、高效和接近汇编语言等特点
  - 对C的类型系统进行了改革和扩充
  - C++也支持面向过程的程序设计，不是一个纯正的面向对象的语言
- 支持面向对象的方法

53

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

函数体开始

主函数

分号，一条完整语句的结束符

函数体结束

输出流，在屏幕上打印引号内的字符串

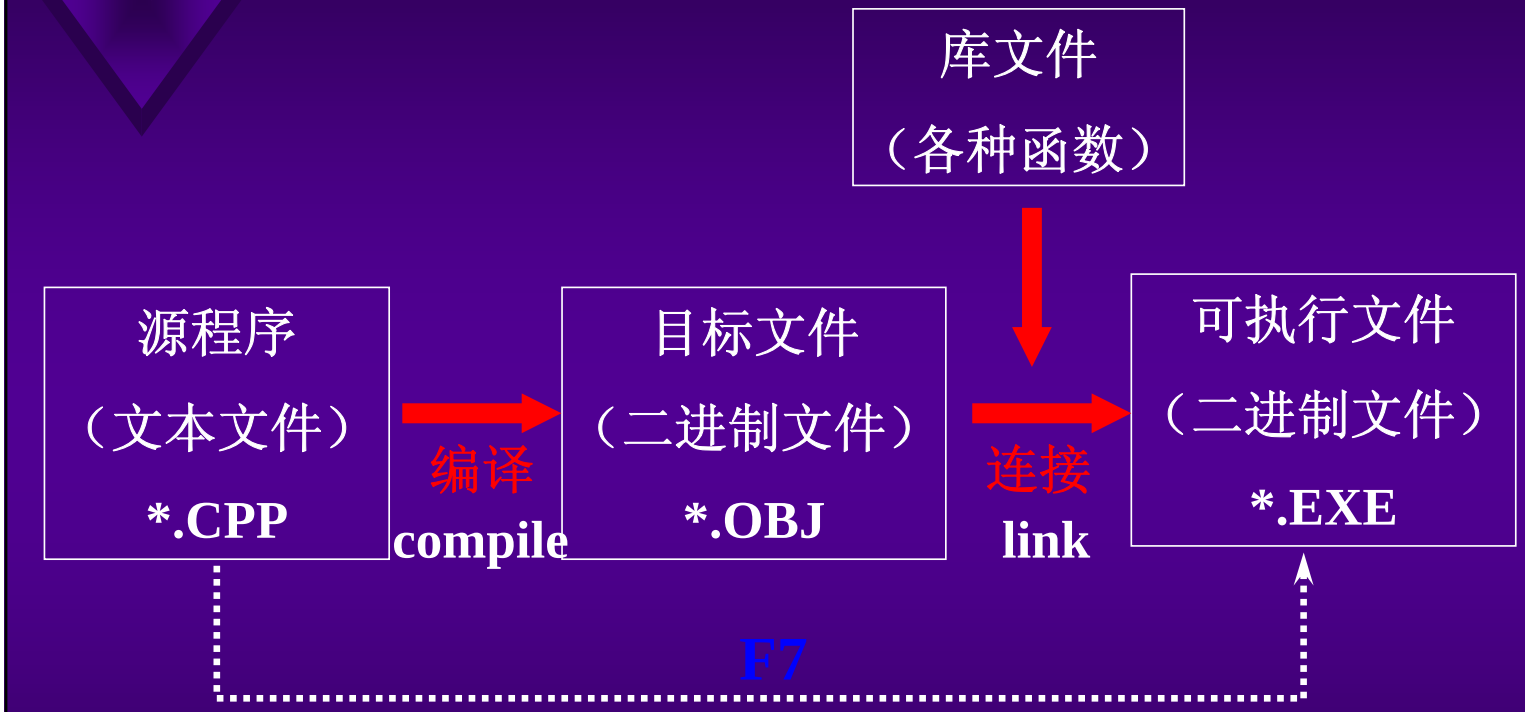
注释或说明

本程序编译执行后，在DOS屏幕上打印出

```
I  am  a student.
```


54

C++程序的编写和实现

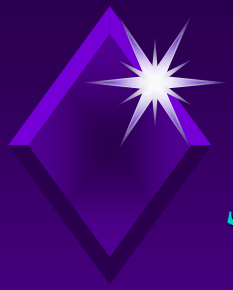

库文件
（各种函数）
源程序
（文本文件）
*.CPP
编译
compile
目标文件
（二进制文件）
*.OBJ
连接
link
可执行文件
（二进制文件）
*.EXE
F7

运行C++程序的步骤和方法

上机运行C++程序的方法

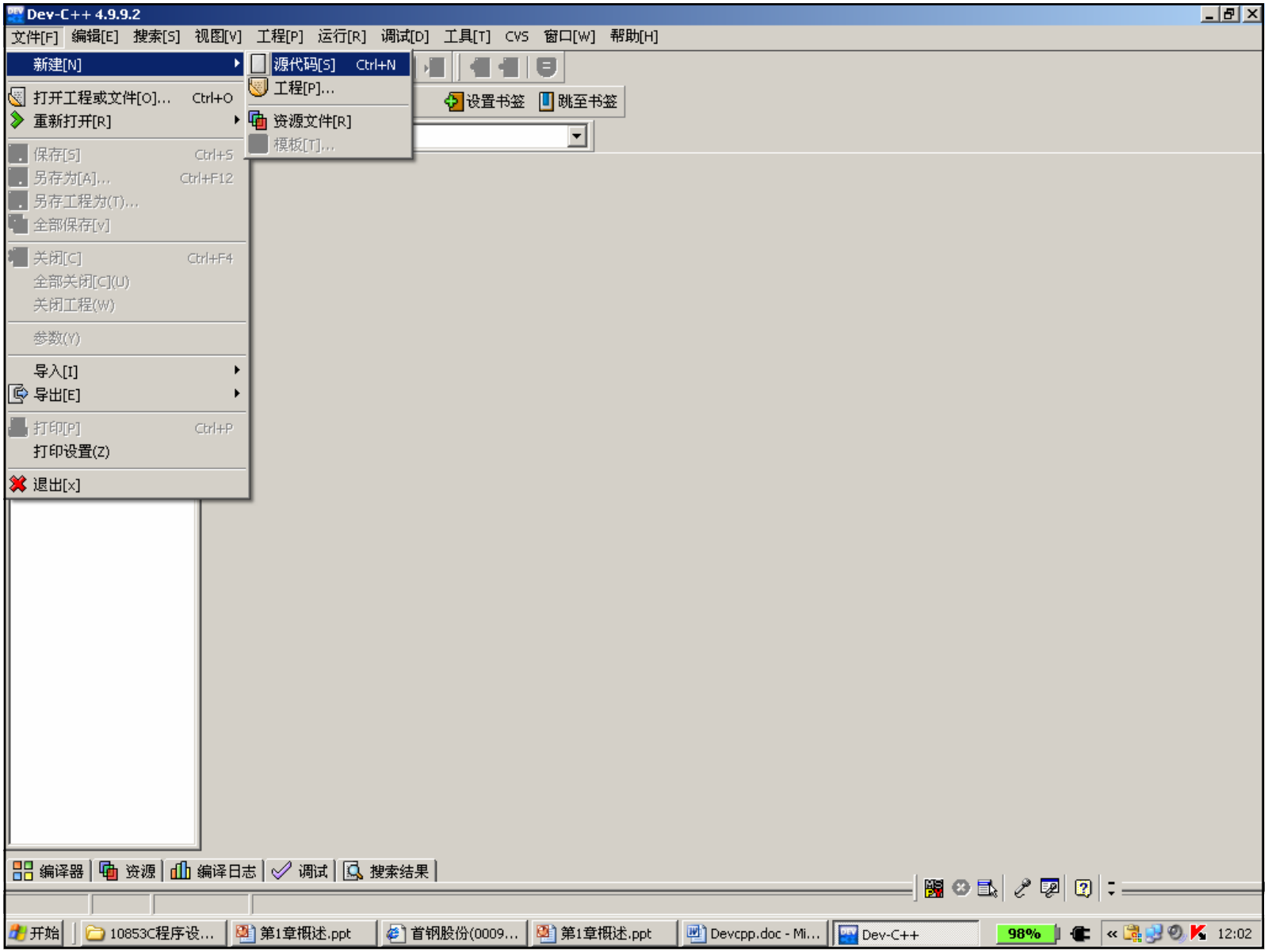
- 目前使用的大多数C++编译系统都是集成环境(IDE)的。可以用不同的编译系统对C++程序进行操作。
- 常用的有Turbo C++ 3.0、Visual C++6.0、Dev5.0等。
- Turbo C++ 3.0: 是一个集成环境，它具有方便、直观和易用的界面，虽然它也是DOS环境下的集成环境，但是可以把启动Turbo C++ 3.0 集成环境的DOS执行文件tc.exe生成快捷方式，也可以用鼠标操作。
- Visual C++6.0(Visual Studio): 也可以用Visual C++6.0对C++程序进行编译。

56

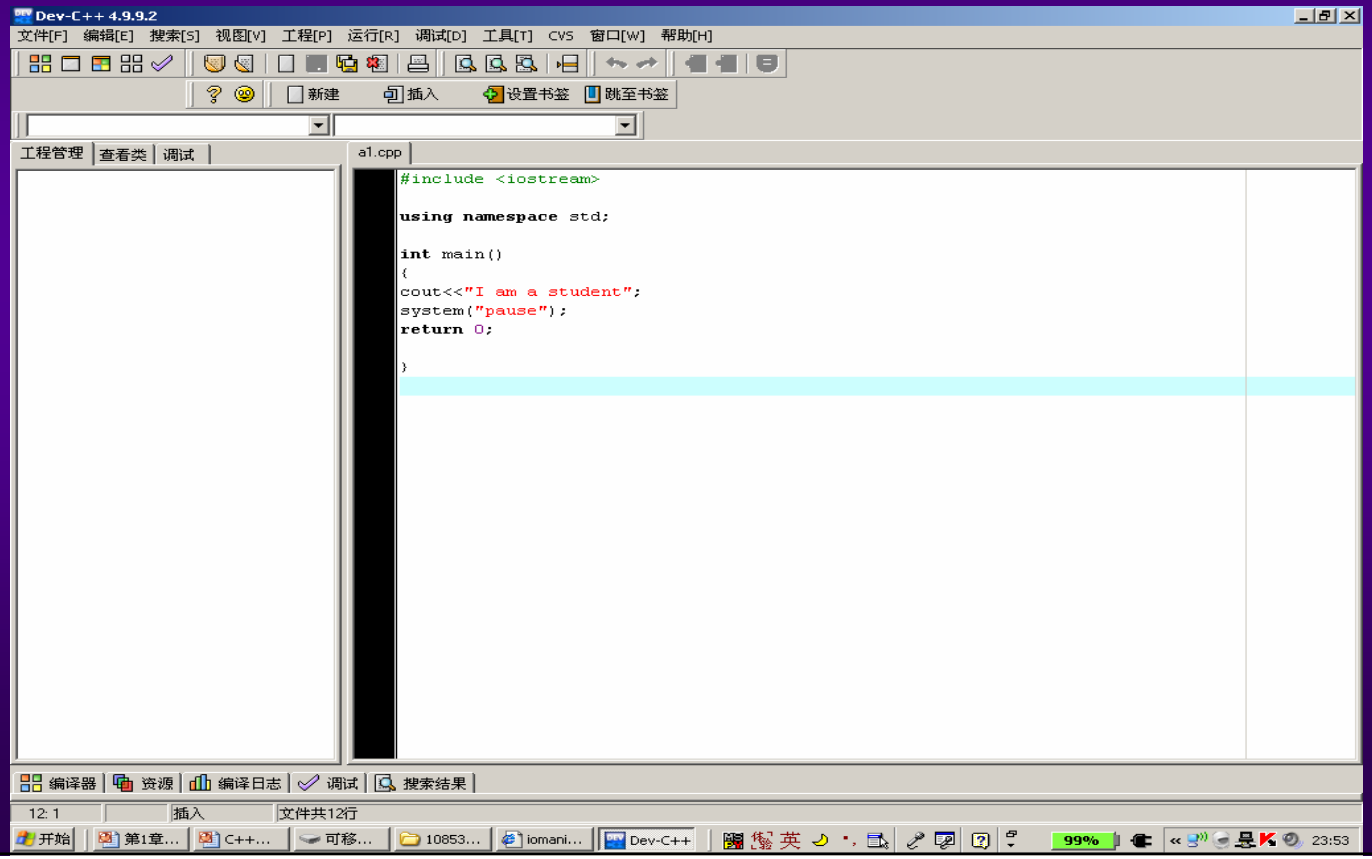


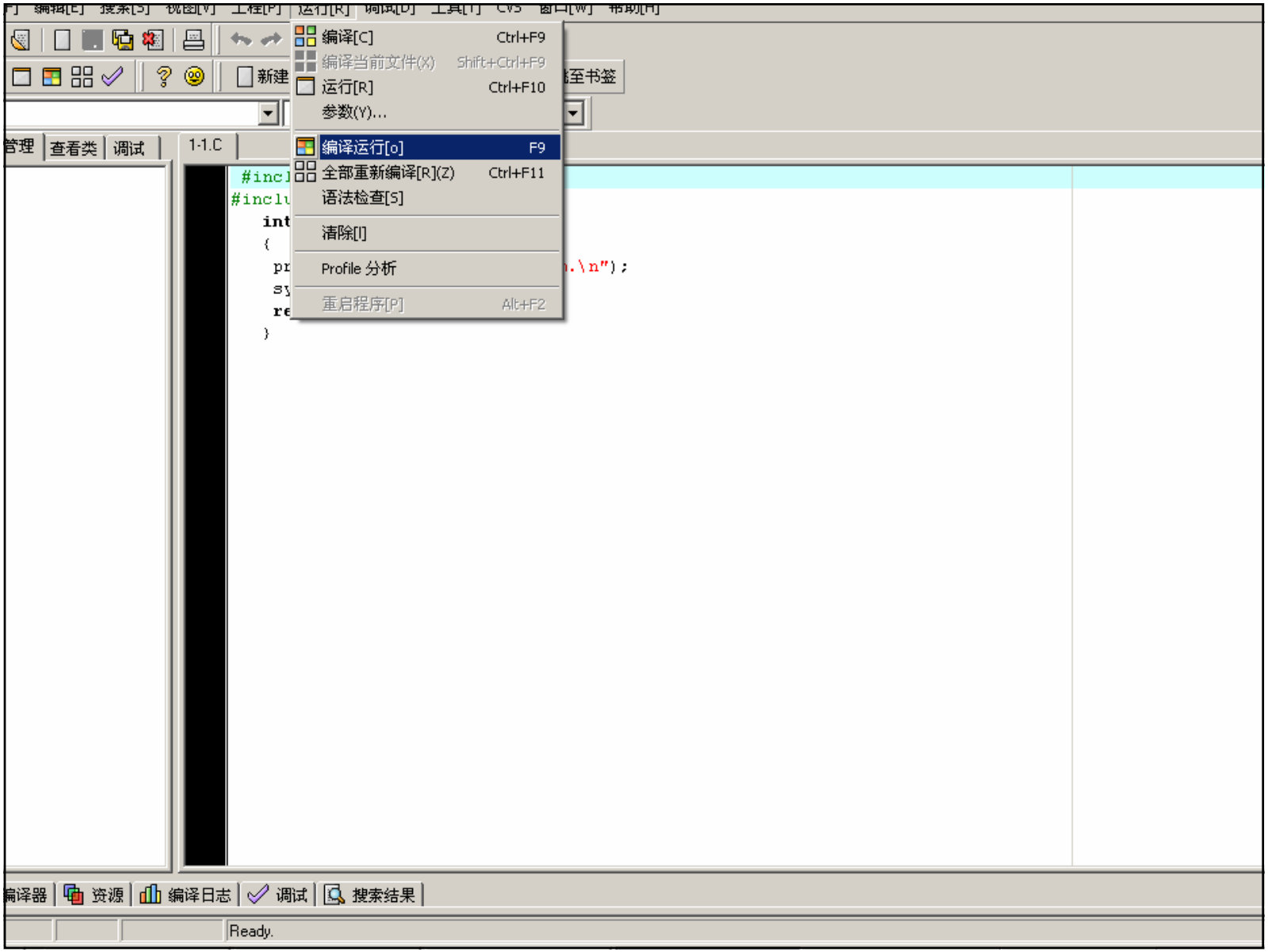
使用Dev C++ 4.9.9.2(也称Dev C++ 5)开发环境

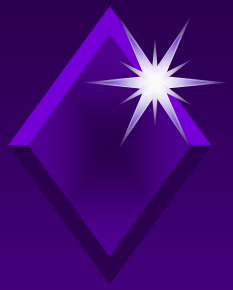
- Dev-C++是一个Windows下的C和C++程序的集成开发环境。


Dev-C++ 4.9.9.2
文件[F] 编辑[E] 搜索[S] 视图[V] 工程[P] 运行[R] 调试[D] 工具[T] CVS 窗口[W] 帮助[H]
新建[N]
打开工程或文件[O]... Ctrl+O
重新打开[R]
保存[S] Ctrl+S
另存为[A]... Ctrl+F12
另存工程为(T)...
全部保存[V]
关闭[C] Ctrl+F4
全部关闭[C](U)
关闭工程(W)
参数(Y)
导入[I]
导出[E]
打印[P] Ctrl+P
打印设置(Z)
退出[x]
源代码[S] Ctrl+N
工程[P]...
资源文件[R]
模板[T]...
设置书签
跳至书签
编译器
资源
编译日志
调试
搜索结果
开始
10853C程序设...
第1章概述.ppt
首钢股份(0009...
第1章概述.ppt
Devcpp.doc - Mi...
Dev-C++
98%
12:02

编写完成后按运行-〉编译运行，或按f9




编译[C] Ctrl+F9
编译当前文件(X) Shift+Ctrl+F9
运行[R] Ctrl+F10
参数(Y)...
编译运行[o] F9
全部重新编译[R](Z) Ctrl+F11
语法检查[S]
清除[I]
Profile 分析
重启程序[P] Alt+F2
管理
查看类
调试
1-1.C
编译器
资源
编译日志
调试
搜索结果
Ready.



- 写完C++程序以后运行的时候,窗口一闪而过。解决办法是在main函数的return0之前添加一句 system("pause"); 来中断程序。

61

使用Visual C++6.0：

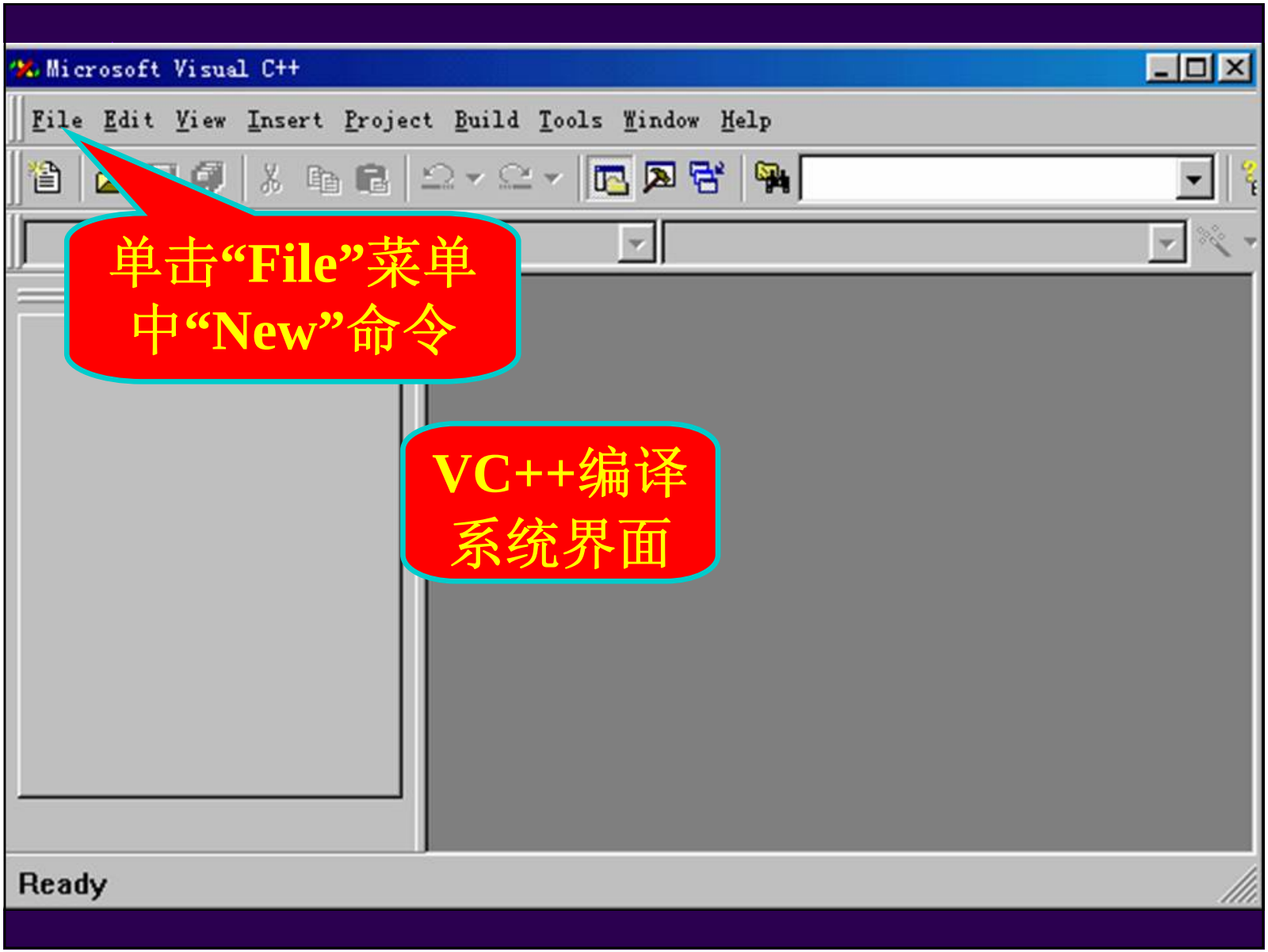
1）启动Visual C++,选择“文件”菜单中的“新建”命令，选择“文件”标签中的“C++ Source File”选项。

2）选择源程序存放的目录和输入源程序名，单击“确定”。

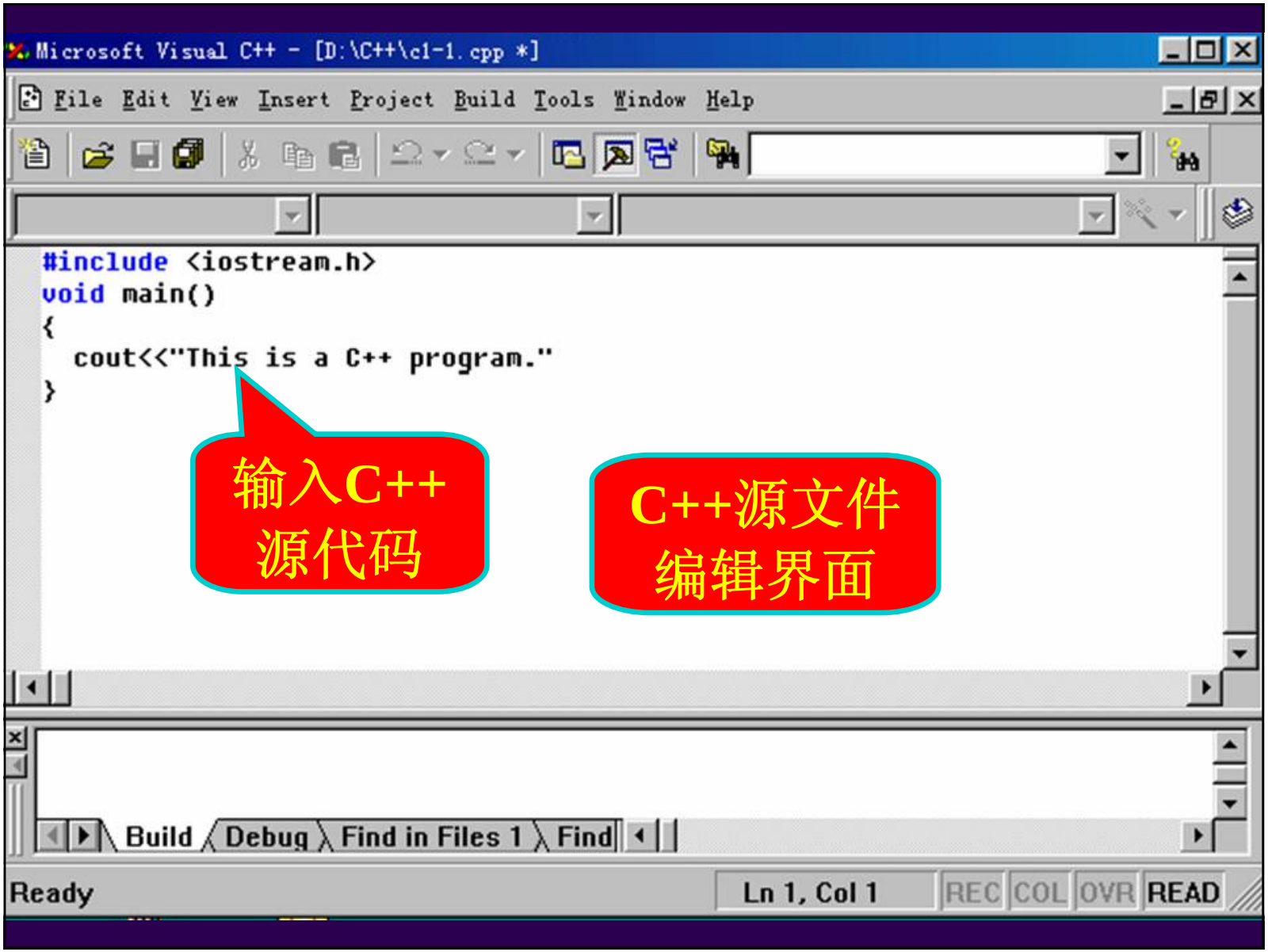
3）在编辑器中编写源程序。

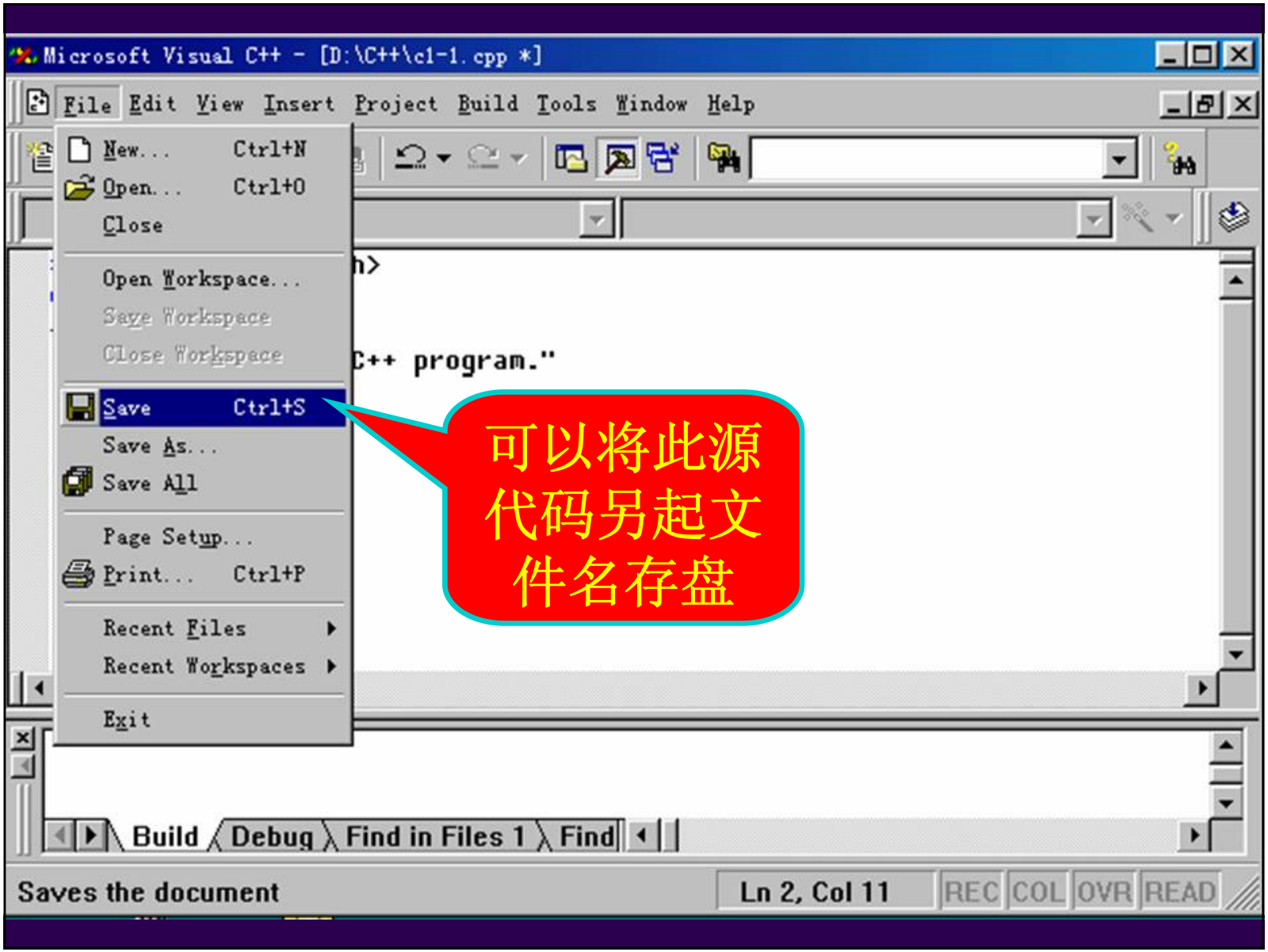
4）单击F7或“编译”中的“重建全部”编译源程序，若编译通过，单击“执行”，在DOS屏上看结果，任按一键返回编辑器。


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
启动VC++编译系统
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

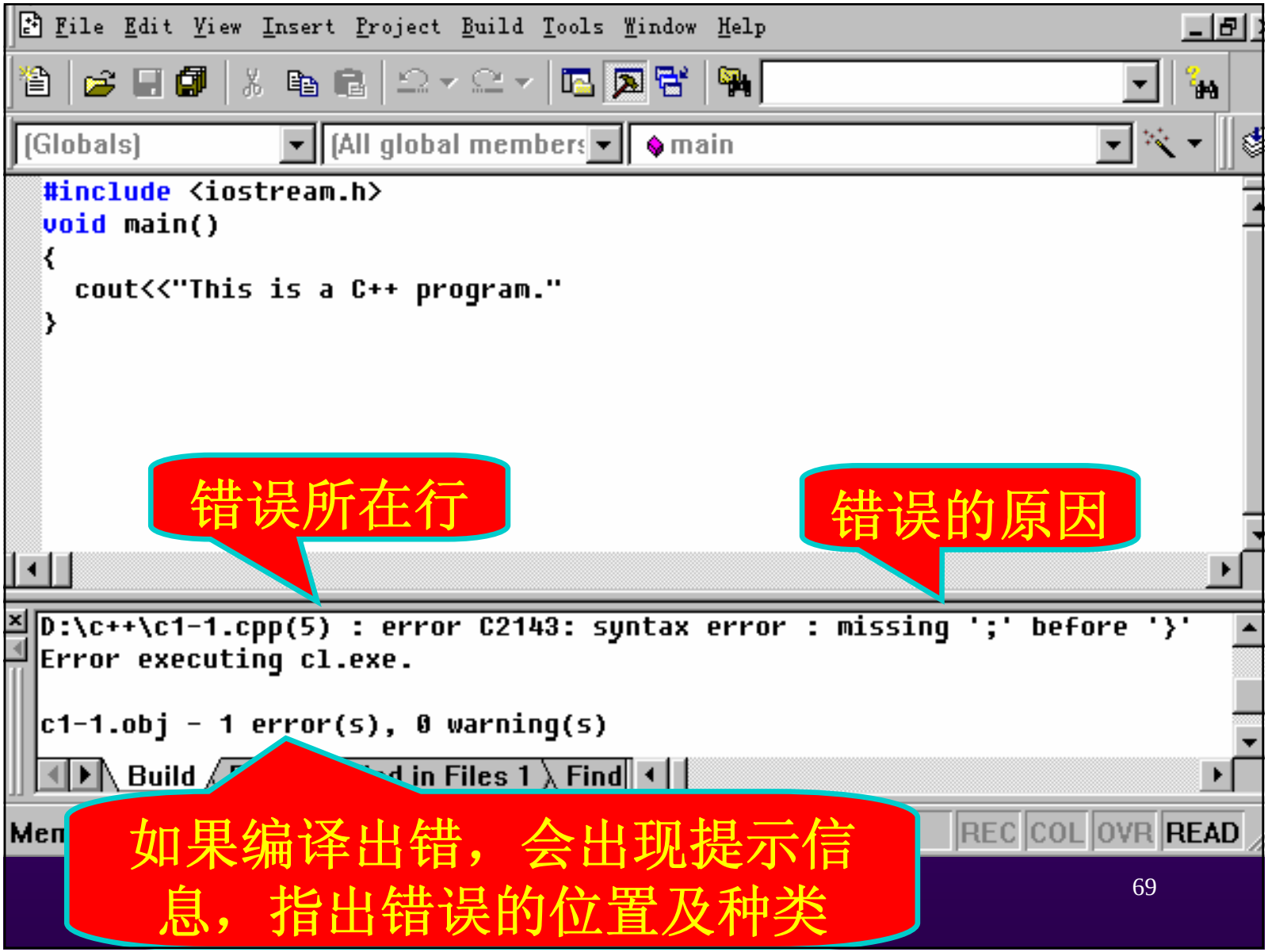

Microsoft Visual C++
File Edit View Insert Project Build Tools Window Help
单击“File”菜单
中“New”命令
VC++编译
系统界面
Ready


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:
确定
结束
网络...
C:\Documents and Settings\zhu
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

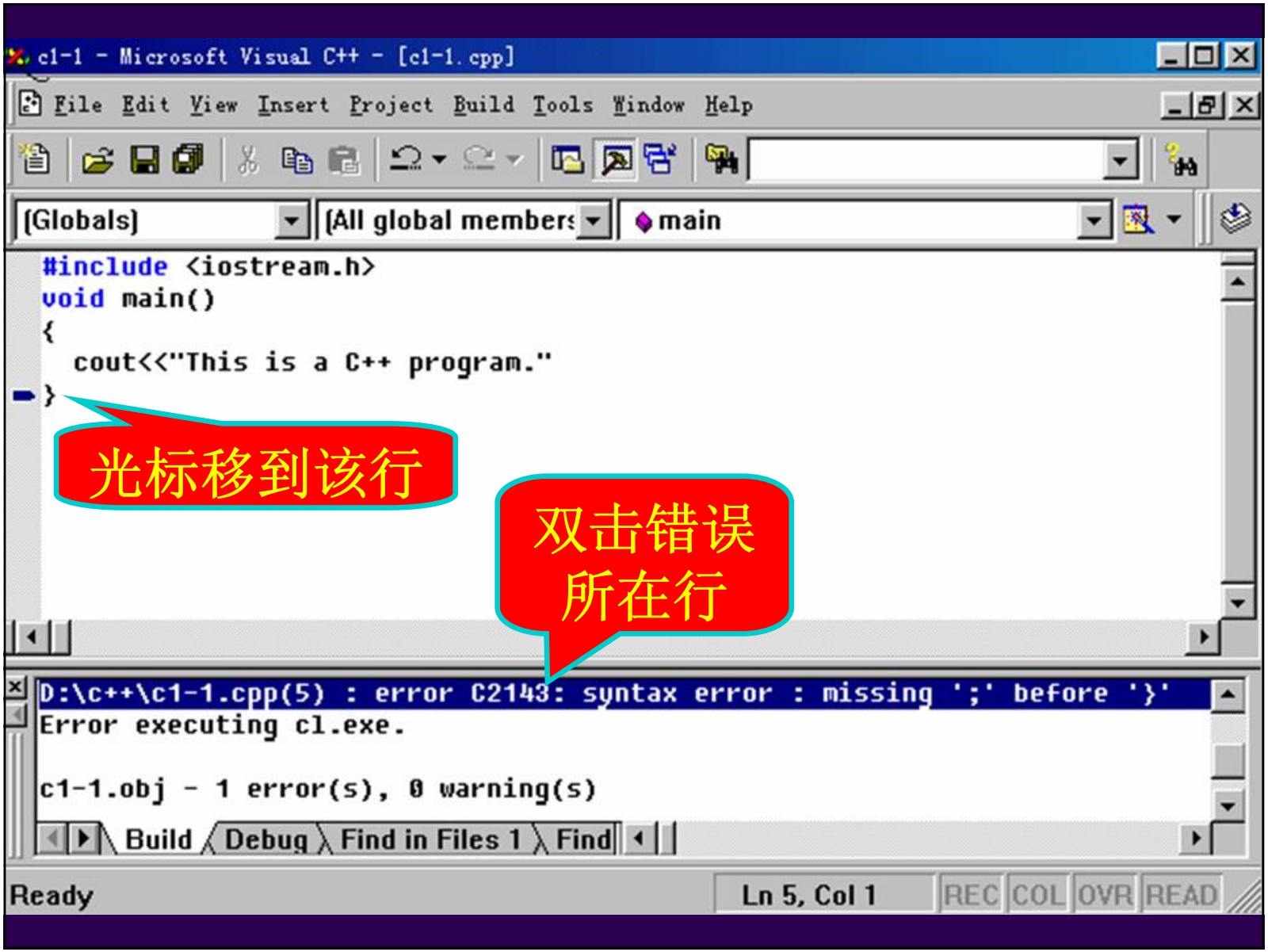

Microsoft Visual C++ - [D:\C++\c1-1.cpp *]
File Edit View Insert Project Build Tools Window Help
#include <iostream.h>
void main()
{
cout<<"This is a C++ program."
}
输入C++
源代码
C++源文件
编辑界面
Build Debug Find in Files 1 Find
Ready
Ln 1, Col 1
REC COL OVR READ

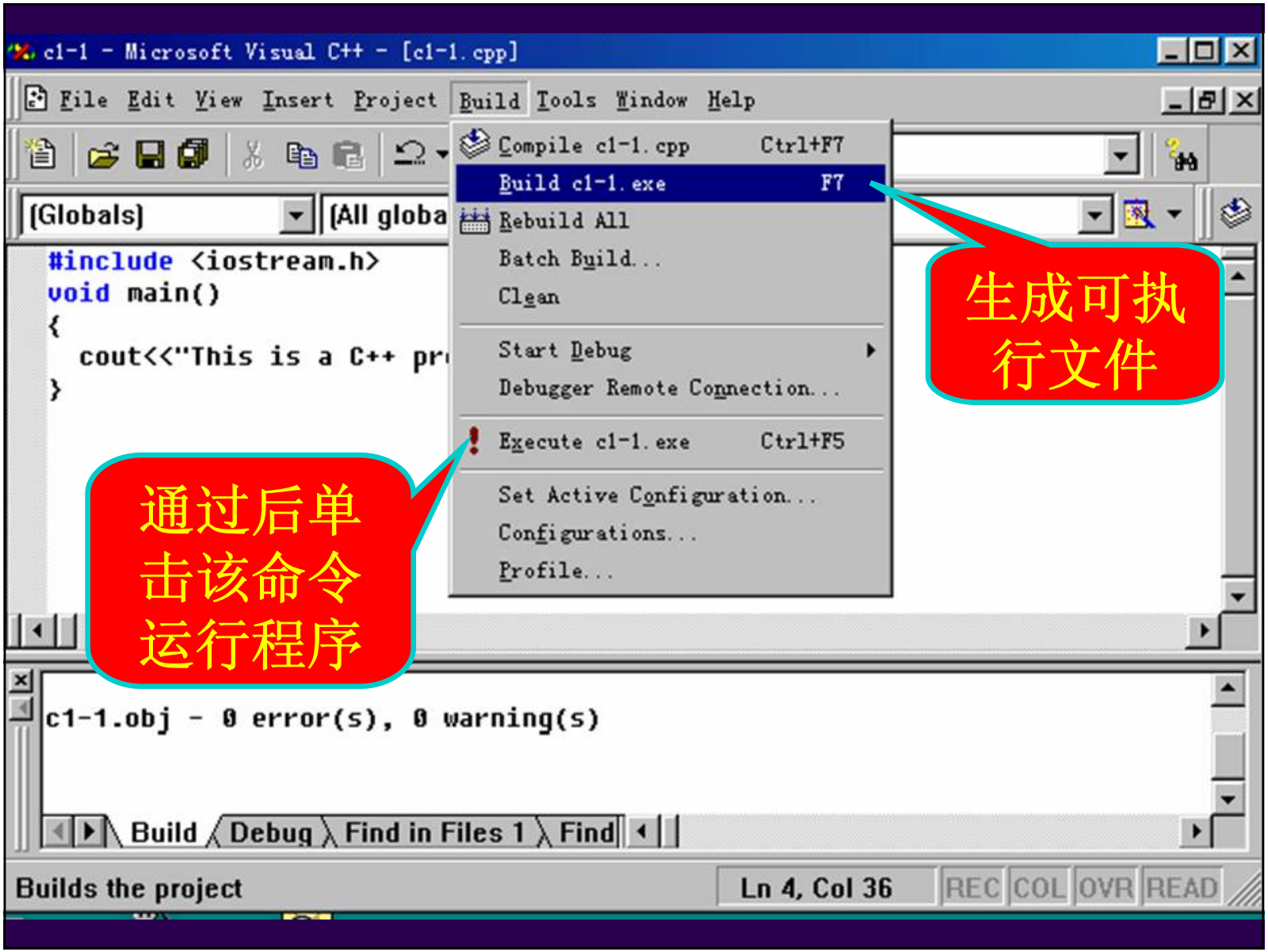

Microsoft Visual C++ - [D:\C++\c1-1.cpp *]
File Edit View Insert Project Build Tools Window Help
New... Ctrl+N
Open... Ctrl+O
Close
Open Workspace...
Save Workspace
Close Workspace
Save Ctrl+S
Save As...
Save All
Page Setup...
Print... Ctrl+P
Recent Files
Recent Workspaces
Exit
h>
C++ program."
可以将此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

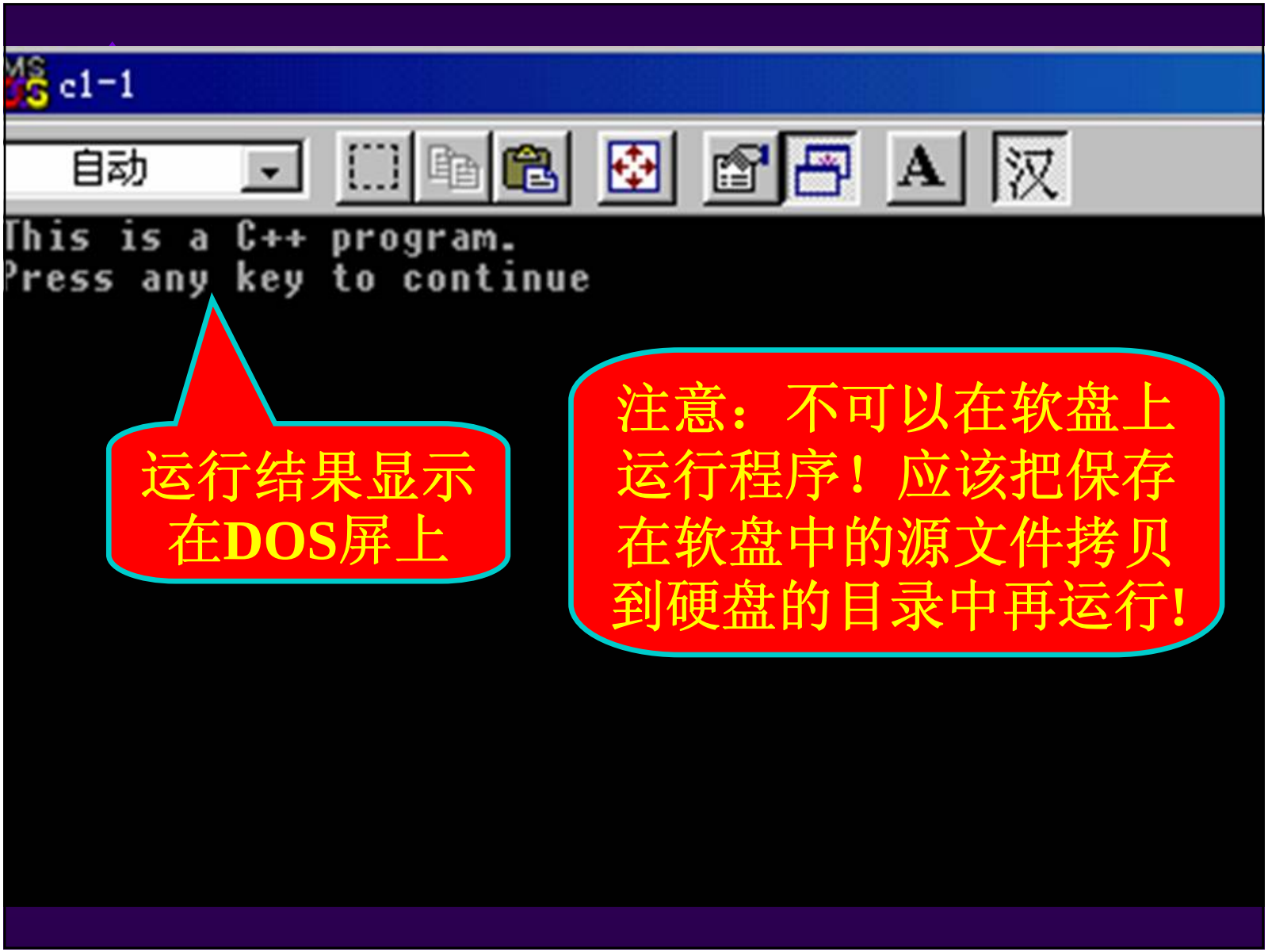

选择编译命令，将源文件.cpp生成.obj文件
Microsoft Visual C++ - [D:\C++\c1-1.
File Edit View Insert Project Build Tools Window Help
Compile c1-1.cpp
Build
Rebuild All
Start Debug
Debugger Remote Connection...
#include <iostream.h>
void main()
{
cout<<"This is a C++ program."
}
68

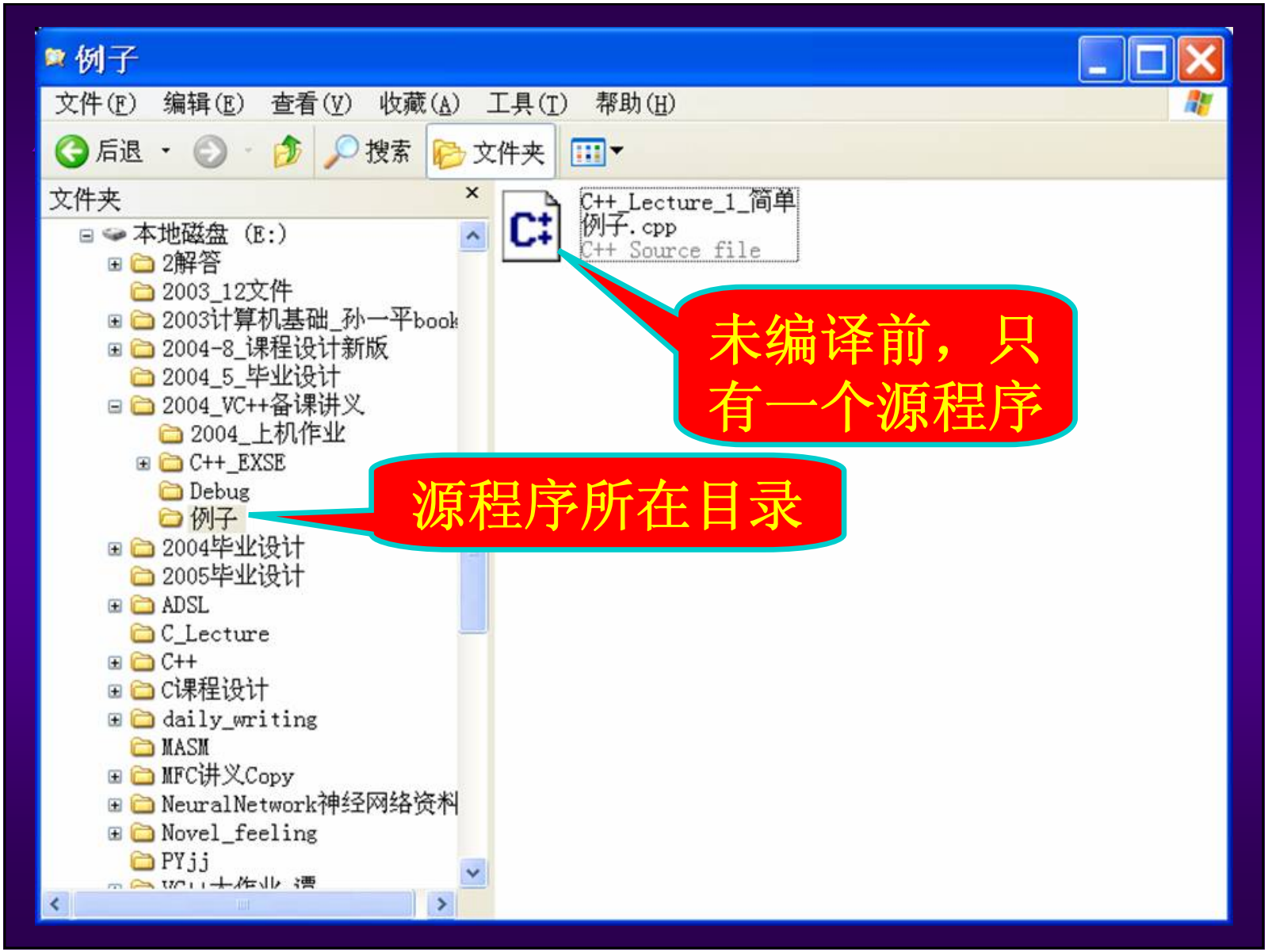

File Edit View Insert Project Build Tools Window Help
(Globals)
(All global members
main
#include <iostream.h>
void main()
{
cout<<"This is a C++ program."
}
错误所在行
错误的原因
D:\c++\c1-1.cpp(5) : error C2143: syntax error : missing ';' before '}'
Error executing cl.exe.
c1-1.obj - 1 error(s), 0 warning(s)
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

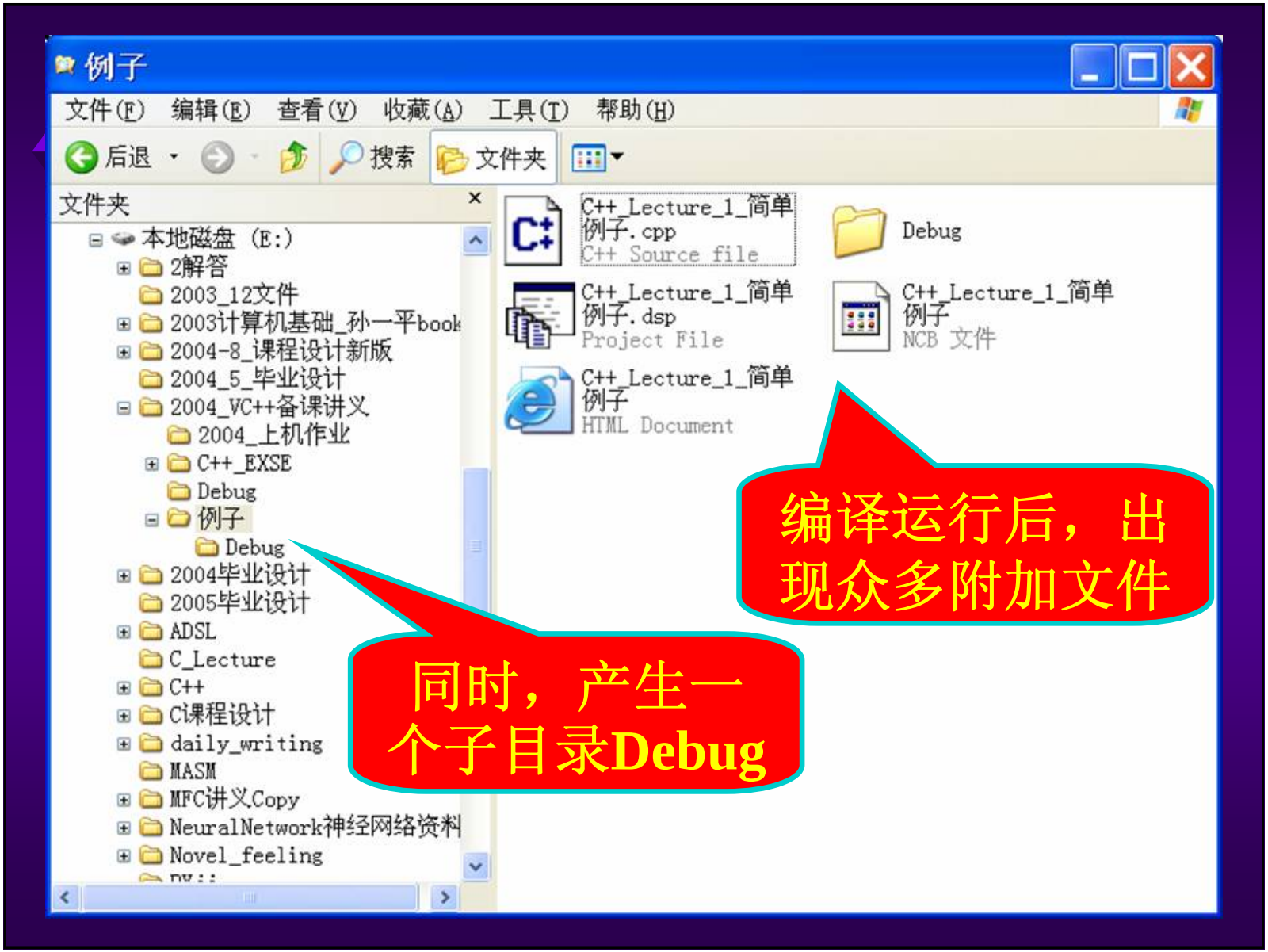
69

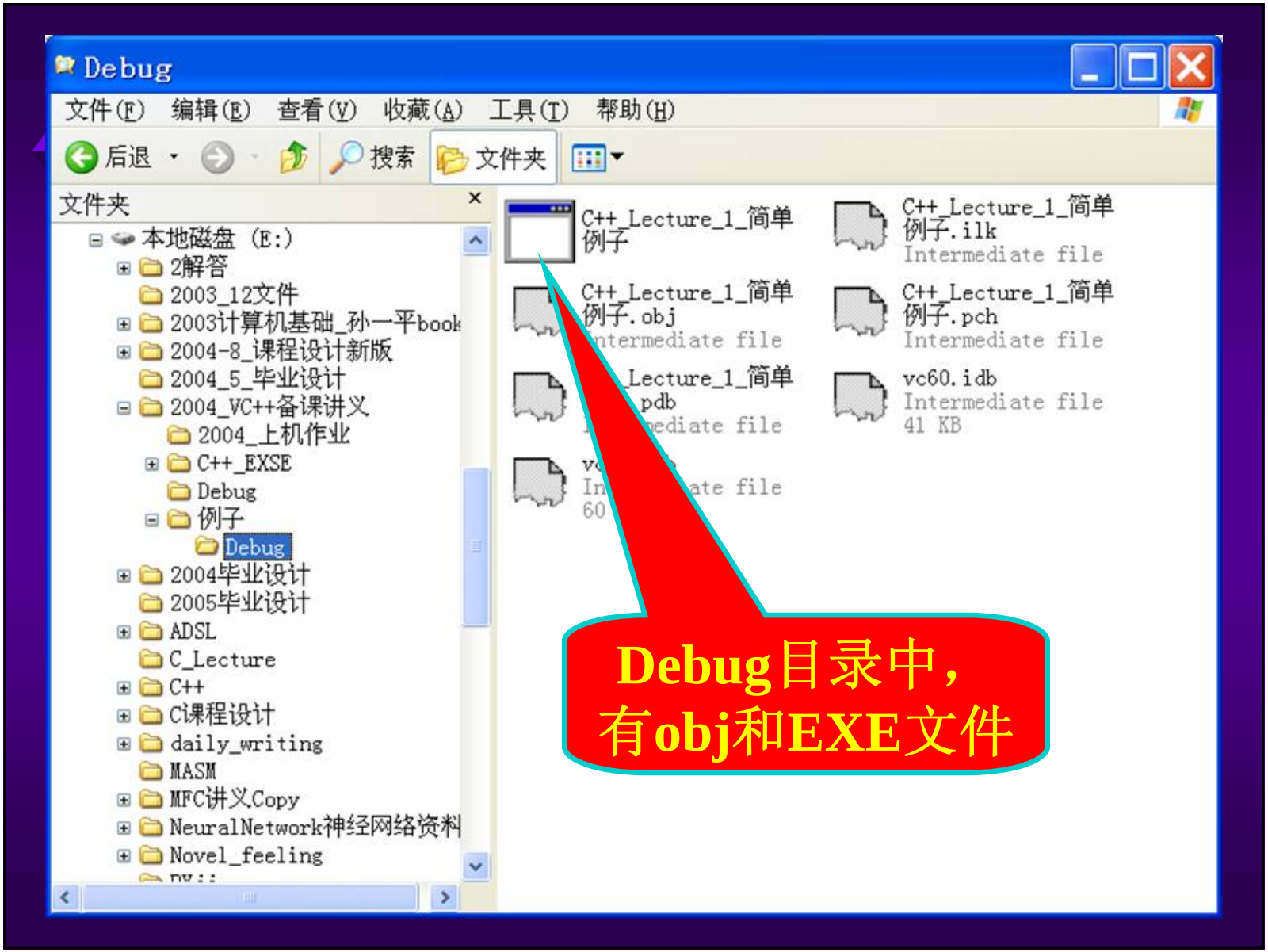

c1-1 - Microsoft Visual C++ - [c1-1.cpp]
File Edit View Insert Project Build Tools Window Help
(Globals)
(All global members
main
#include <iostream.h>
void main()
{
cout<<"This is a C++ program."
}
光标移到该行
双击错误
所在行
D:\c++\c1-1.cpp(5) : error C2143: syntax error : missing ';' before '}'
Error executing 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)
(All globa
#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：不可以在软盘上运行程序！应该把保存在软盘中的源文件拷贝到硬盘的目录中再运行!


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
C++_Lecture_1_简单例子.pch
Intermediate file
vc60.idb
Intermediate file
41 KB
Debug目录中，
有obj和EXE文件

另一个例子

```
#include  <iostream.h>
void  main(void)
{
  cout << “i=”;
  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
}
```


76