



第1章：C++语言概述



从 C 语言 到 C++

- 1、C语言是一种结构化的程序设计语言，**语言本身简洁、使用灵活方便**。既适用于设计和编写大的系统程序，又适用于编写小的控制程序，也适用科学计算。
- 2、**它既有高级语言的特点，又具有汇编语言的特点**。运算符丰富，除了提供对数据的算术逻辑运算外，还提供了二进制的位运算。并且也提供了灵活的数据结构。用C语言编写的程序表述灵活方便，功能强大。用C语言开发的程序，其结构性好，目标程序质量高，程序执行效率高。



3、**程序的可移植性好**。用C语言在某一种型号的计算机上开发的程序，基本上可以不作修改，而直接移植到其它型号和不同档次的计算机上运行。

4、**程序的语法结构不够严密，程序设计的自由度大**。往往是编好程序输入计算机后，编译时容易通过，而在执行时还会出错。但只要对C语言的语法规则真正领会，编写程序及调试程序还是比较容易掌握的。



随着C语言应用的推广，C语言存在的一些缺陷或不足也开始流露出来，并受到大家的关注。如：**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；随着软件规模的扩大，难以适应开发特大型的程**度等等。



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。

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）、BC++（Borland C Plus Plus）、AT&T C++等。



面向对象程序设计的有关概念

- 面向对象方法的由来和发展
- 面向对象的基本概念



面向对象方法的由来和发展

面向对象方法是求解问题的一种新方法

把求解问题中客观存在的事物看作各自不同的对象;
再把具有相同特性的一些对象归属为一个类

面向对象方法是计算机科学发展要求

满足了人们对信息的需求量越来越大
对软件开发的规模也越来越大
对软件可靠性和代码的重用性的要求越来越高的客观需要。



面向对象的基本概念

- 对象
- 类
- 封装
- 继承
- 多态性



对象

一般意义上的对象：

对象是现实世界中客观存在的某种事物。

- 自然物体（有形的，汽车、房屋、猫）
- 逻辑结构（无形的，班级、支部、连队）
- 其他（无形的，如一项计划）

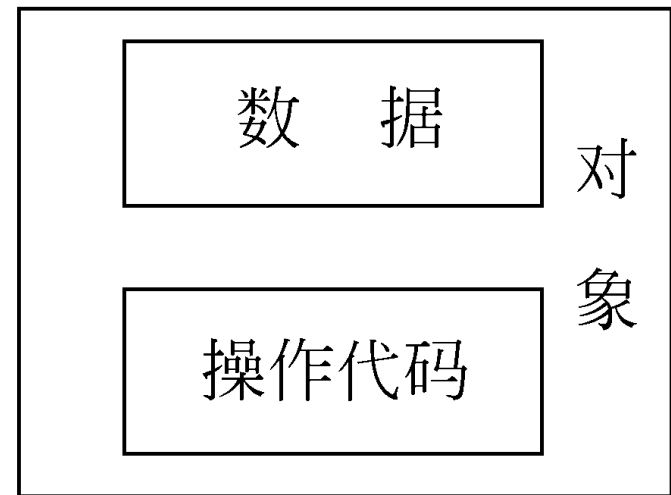
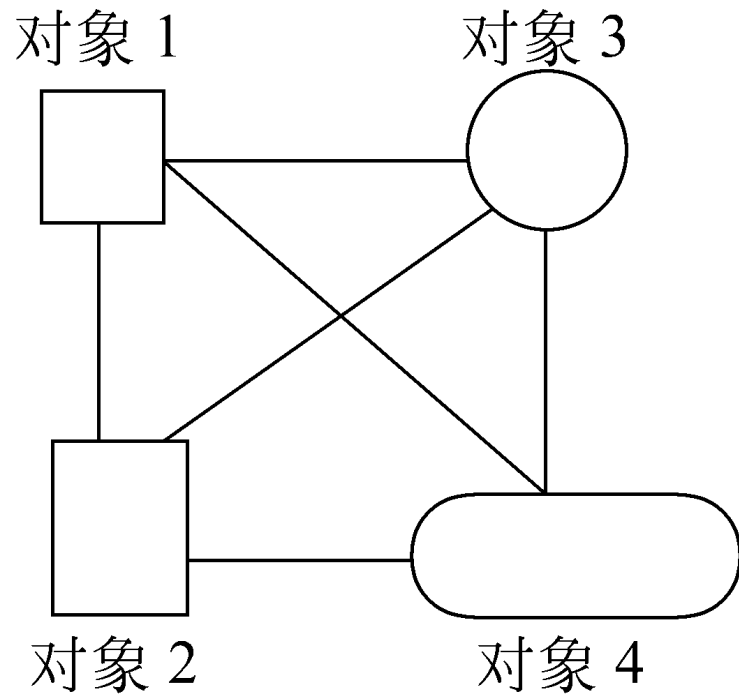
对象是一种相对独立的实体

- 通过一组数据来描述对象的静态特性
- 使用一组行为或功能来表示对象的动态特性。



面向对象方法中的对象:

- 是系统中用来描述客观事物的一个实体。
- 对象是构成系统的一个基本单位。
- 对象具有两个要素
 - 属性：用来描述对象静态特征的数据项。
 - 行为：用来描述对象动态特征的操作代码。





类

- 分类——人类通常的思维方法
- 分类所依据的原则——抽象
- 抽象的过程是将有关事物的共性归纳、集中的过程。
- 抽象的作用是表示同一类事物的本质。
- 例如，石头、树木、汽车、房屋等都是人们在长期的生产和生活实践中抽象出的概念。



■ 面向对象方法中的"类"

- 具有相同属性和行为的一组对象的集合
- 为属于该类的全部对象提供了抽象的描述，包括属性和行为两个主要部分。

■ 类与对象的关系：

犹如模具与铸件之间的关系，一个属于某类的对象称为该类的一个实例。



封装

- 封装是指把对象的属性和行为结合成一个独立的单位，又称为封装体。
- 封装体具有独立性和隐藏性。
 - 一是将有关的数据和操作代码封装在一个对象中，形成一个基本单位，各个对象之间相对独立，互不干扰。
 - 二是将对象中某些部分对外隐蔽，即隐蔽其内部细节，只留下少量接口，以便与外界联系，接收外界的消息。



继承

- 利用继承可以简化程序设计的步骤。
- 采用继承的方法可以很方便地利用一个已有的类建立一个新的类。这就是常说的“软件重用”(software reusability) 的思想。
- 继承的定义：特殊类的对象拥有其一般类的全部属性与行为，称作特殊类对一般类的继承。
- 例如：将轮船作为一个一般类，客轮便是一个特殊类。



多态性

多态性指的是一种行为对应着多种不同的实现。
在同一个类中，同一种行为可对应着不同的实现。

- 函数重载和运算符重载

同一种行为在一般类和它的各个特殊类中可以有不同的实现。

- 虚函数



C++语言是面向对象的程序设计语言

- C++语言对面向对象程序设计方法的支持
- C++语言继承了C语言
- C++语言对C语言进行了改进



C++语言对面向对象程序设计方法的支持

1. 支持封装性

- C++语言允许使用类和对象。类是支持数据封装的工具，对象是数据封装的实现。
- 类中成员有不同的访问权限。

2. 支持继承性

- C++语言支持面向对象方法中的继承性，它不仅支持单重继承，而且支持多重继承。

继承和封装是衡量一种语言是否是面向对象的程序设计语言的两个重要指标

3. 支持多态性

- 多态性是在继承性基础上的面向对象方法中的重要特性之一。
 - ① 支持函数重载和运算符重载。
 - ② 支持虚函数。



C++语言继承了C语言

C++语言与C语言兼容，**C语言是C++语言的一个子集。**

C++语言具有C语言的简练明了的风格，同时还保留某些C语言的面向过程的特性。实际上，C++语言是一种不完全的面向对象的程序设计语言。



C++语言对C语言进行了改进

- ① C++语言中规定函数定义时必须指出类型。
- ② C++语言规定函数说明必须使用原型说明，不得用简单说明。
- ③ C++语言规定凡是从高类型向低类型转换时都需加强制转换。
- ④ C++语言中符号常量建议使用const关键字来定义。
- ⑤ C++语言中引进了内联函数。



- ⑥ C++语言允许设置函数参数的默认值。
- ⑦ C++语言引进了函数重载和运算符重载。
- ⑧ C++语言引进了引用概念，使用引用作函数的参数和返回值。
- ⑨ C++语言提供了与C语言不同的I/O流类库，方便了输入/输出操作。
- ⑩ C++语言为方便操作还采取了其他措施。(如 new delete)



C++程序结构的特点

- 简单的输入、输出

向标准输出设备（显示器）输出（插入符）

- 例： `int x;`
- `cout<<"x="<<x;`

从标准输入设备（键盘）输入（提取符）

- 例： `int x;`
- `cin>>x;`



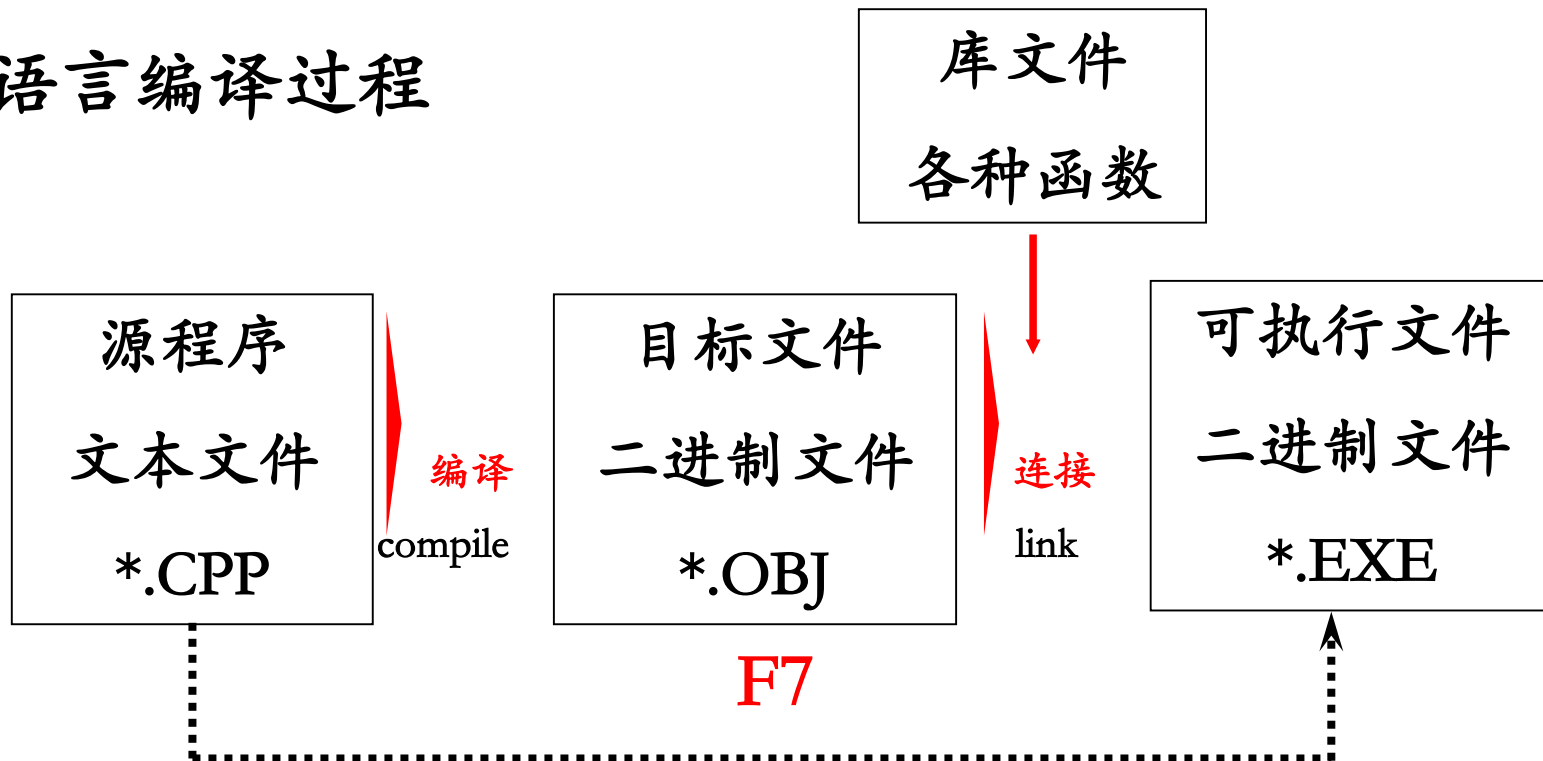
C++程序结构上的特点

- ① C++语言程序是由若干个类和函数组成的。这些类和函数可以放在一个文件中，也可以放在多个文件中。
- ② C++语言程序中的函数有两个种类，一个种类是类体内的成员函数，另一个种类是类体外的一般函数。
- ③ C++程序中有且仅有一个主函数main()。C++程序是从主函数开始执行的。
- ④ C++程序中的函数都是由函数头和函数体构成的，函数体由若干条语句组成的；函数头中包括函数名、函数类型和函数参数。
- ⑤ C++语言程序与C语言程序一样，可以使用预处理命令，也可以使用注释信息。



简单的C++程序介绍

高级语言编译过程



在Vitual 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这两个文件。



一个简单的C++程序

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

包含文件

函数体
开始

主函数

```
void main(void )
```

分号，一条完整
语句的结束符

```
{ cout<<"I am a student.\n"; //输出字符串
```

```
}
```

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

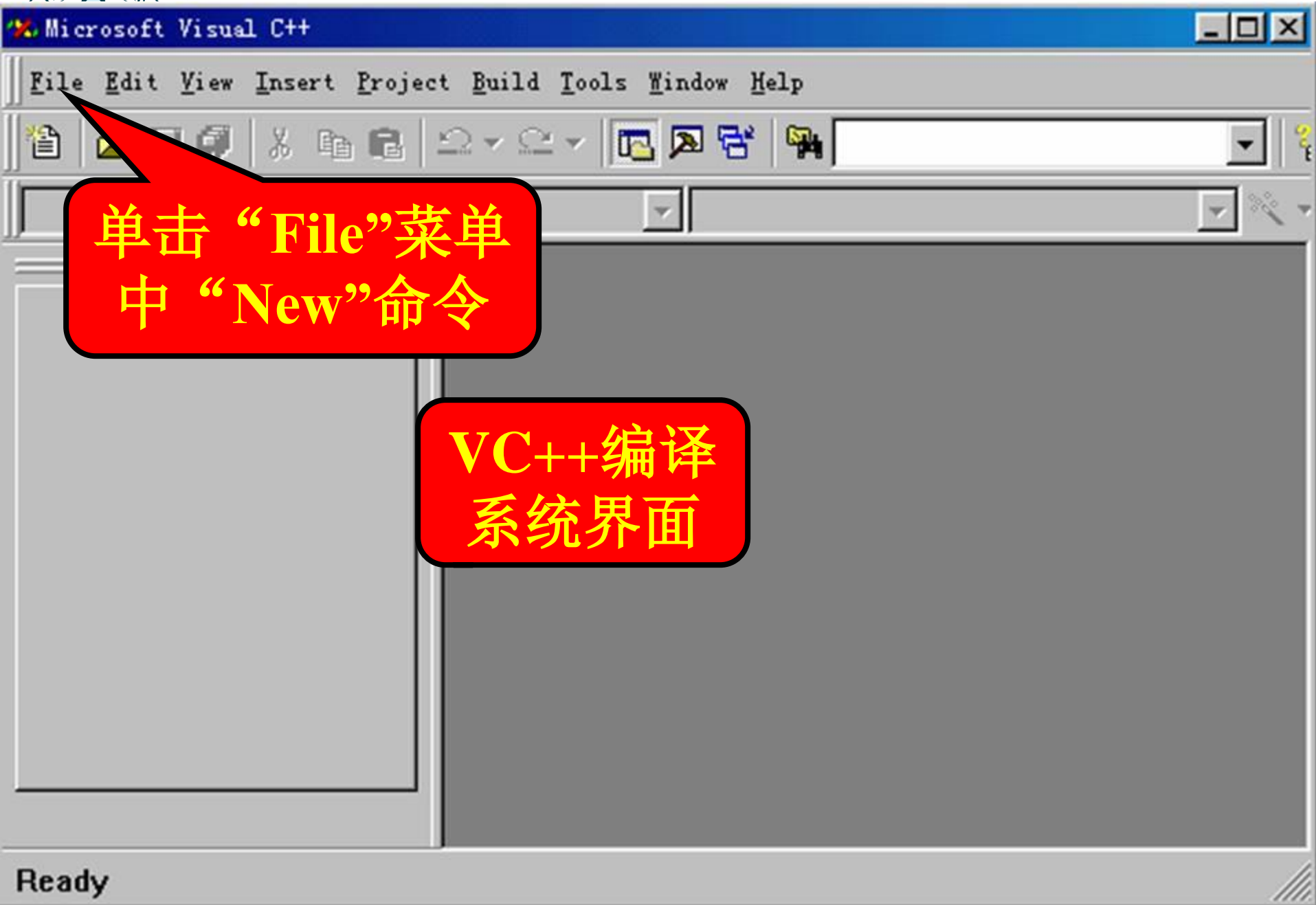
本程序编译执行后，在DOS屏幕上打印出

I am a student.



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。



选择“Files”选项卡

Files Projects Workspaces Other Documents

选择C++源
文件命令

输入文件名

File

c1-1.cpp

Location:

D:\C++

单击选择
驱动器

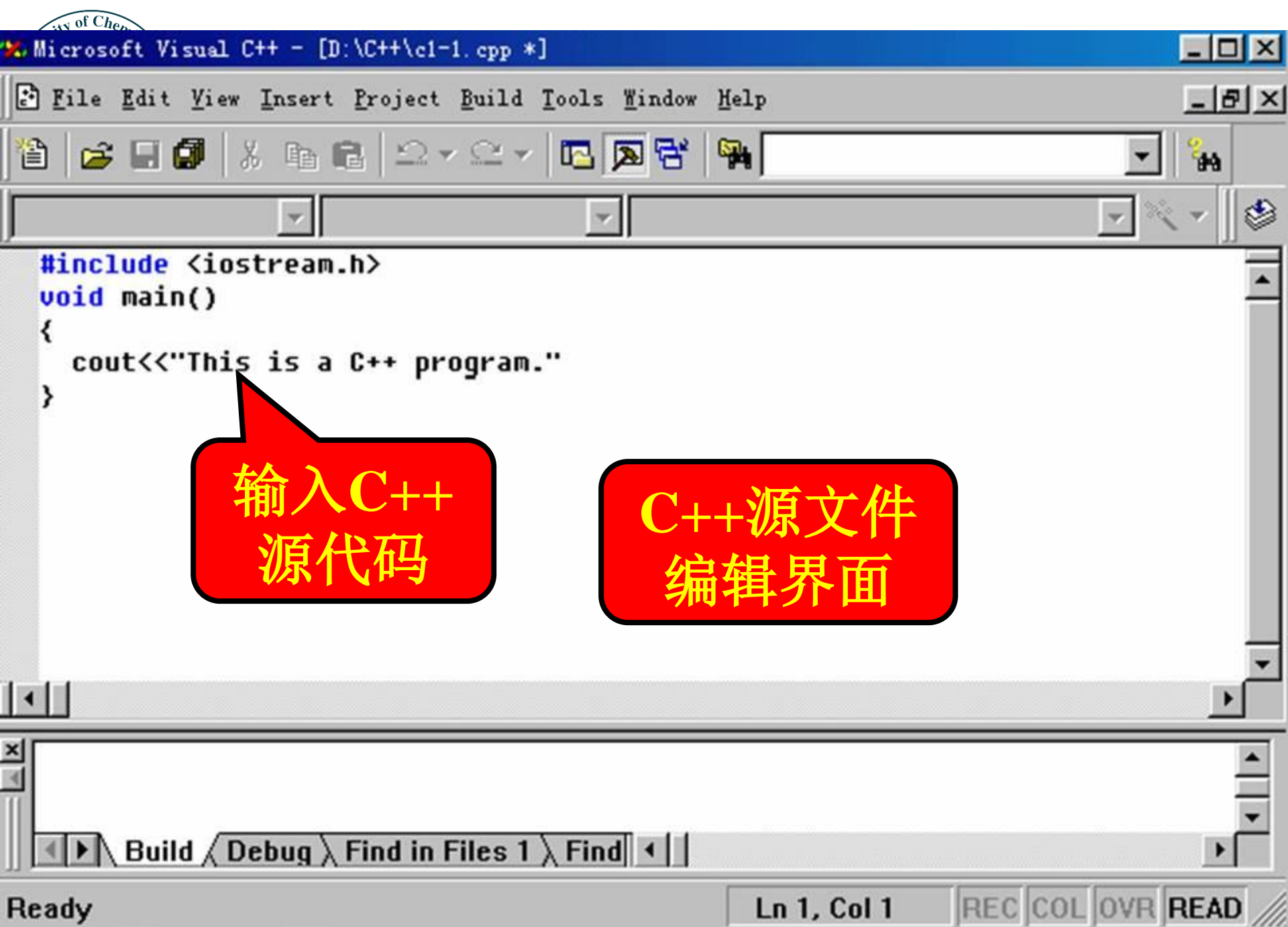
输入文件
存放位置

选择驱动
器或目录

OK

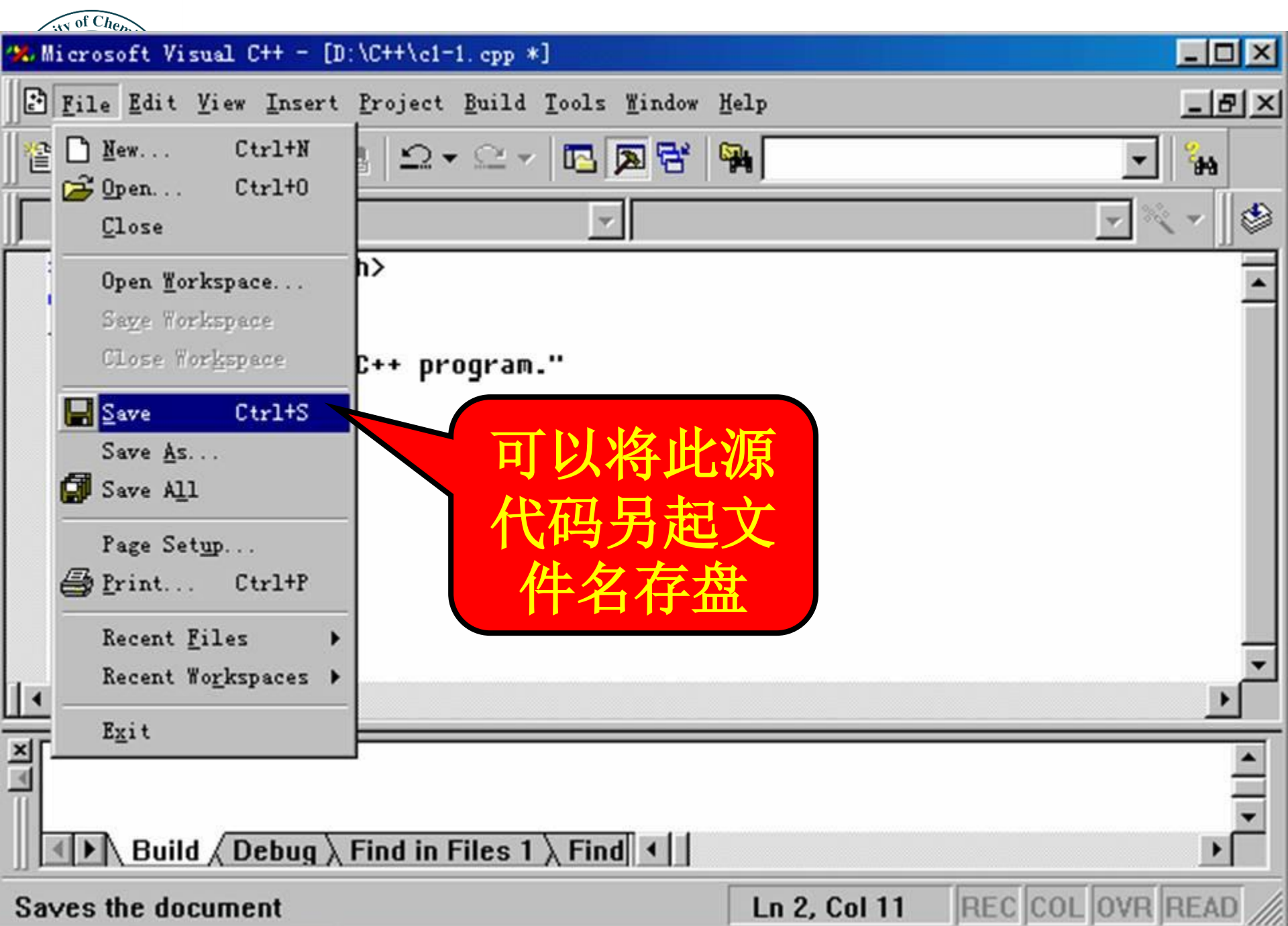
Cancel





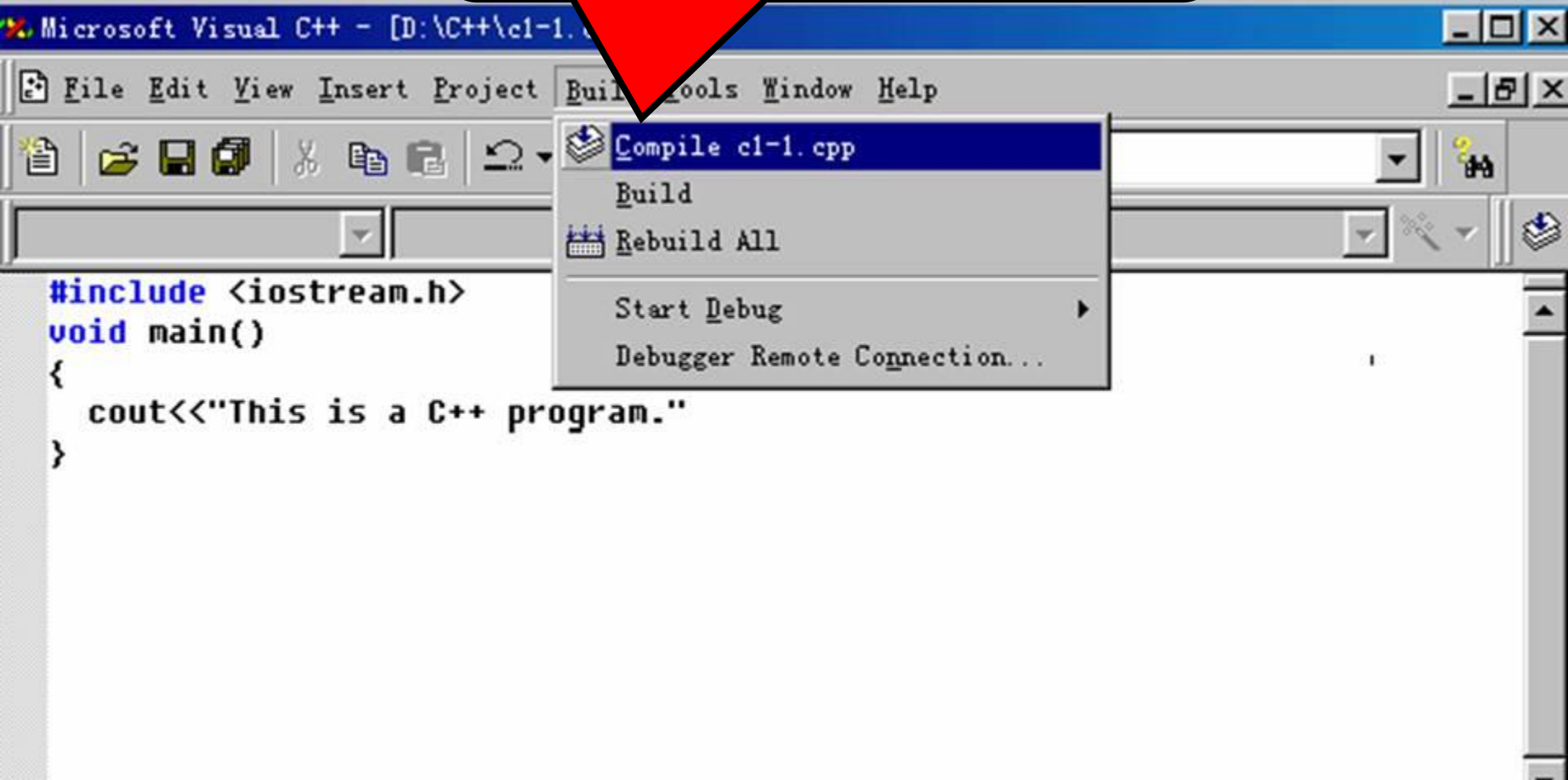
输入C++
源代码

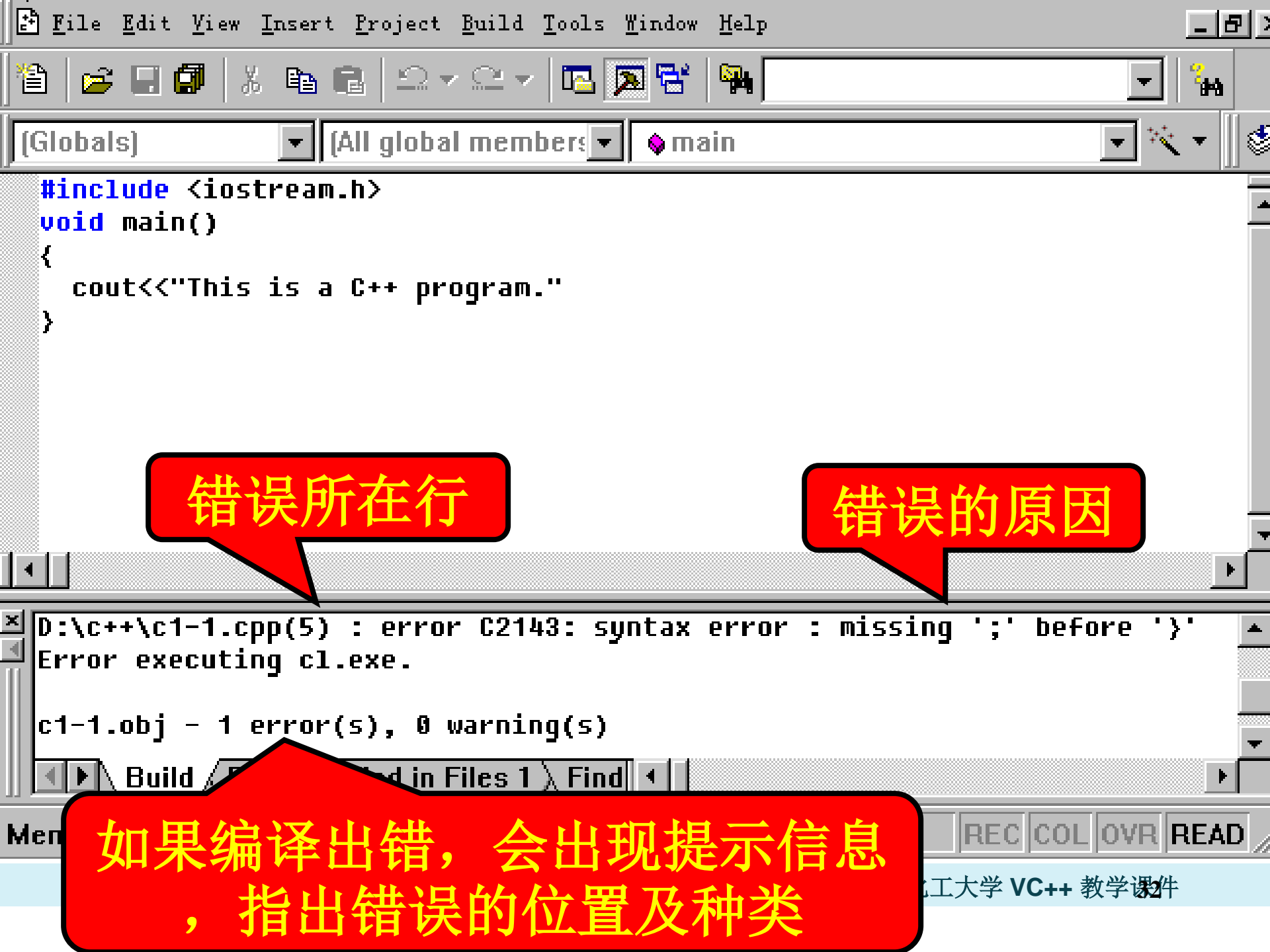
C++源文件
编辑界面





选择编译命令，将源文件.cpp生成.obj文件





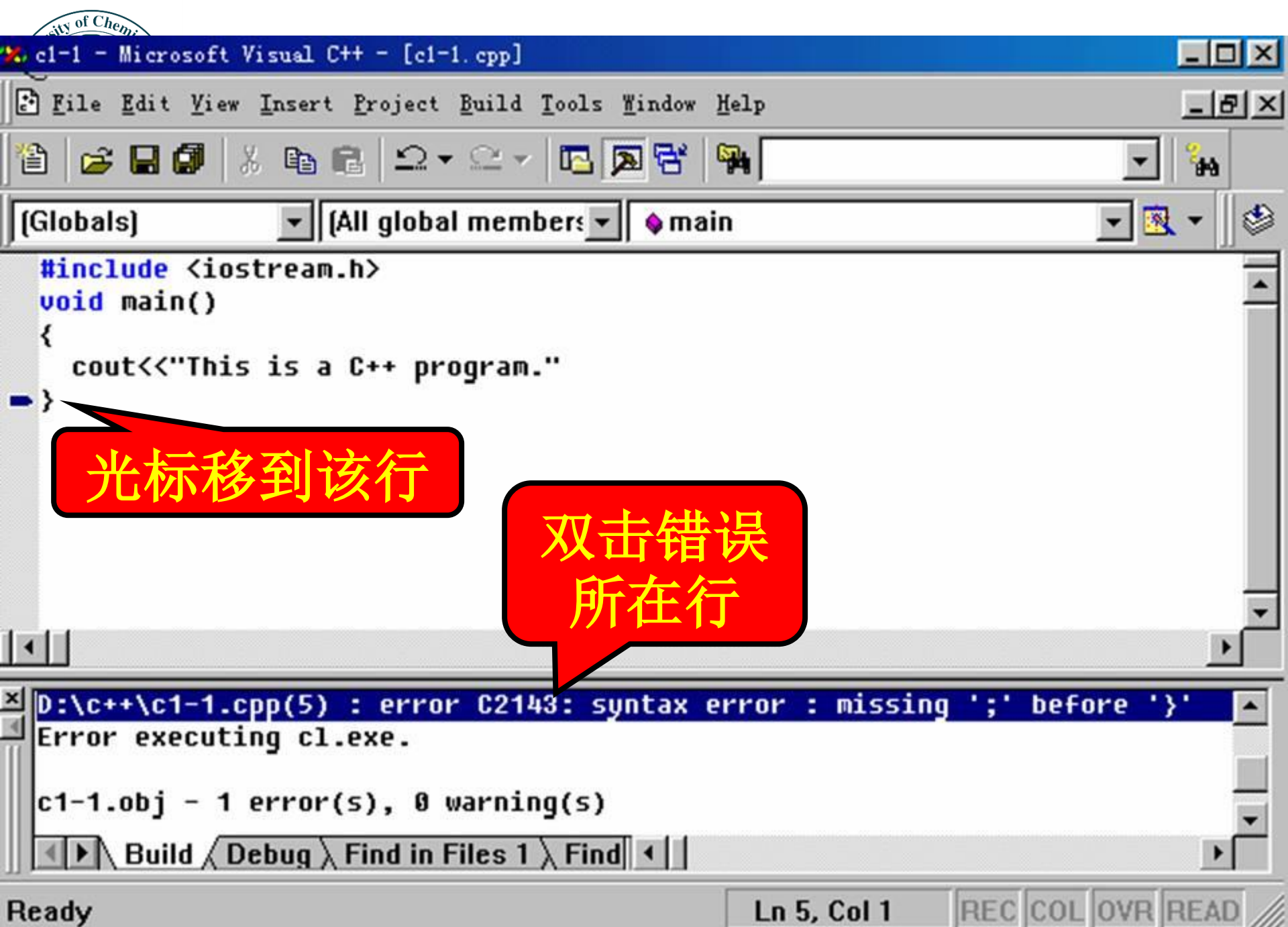
错误所在行

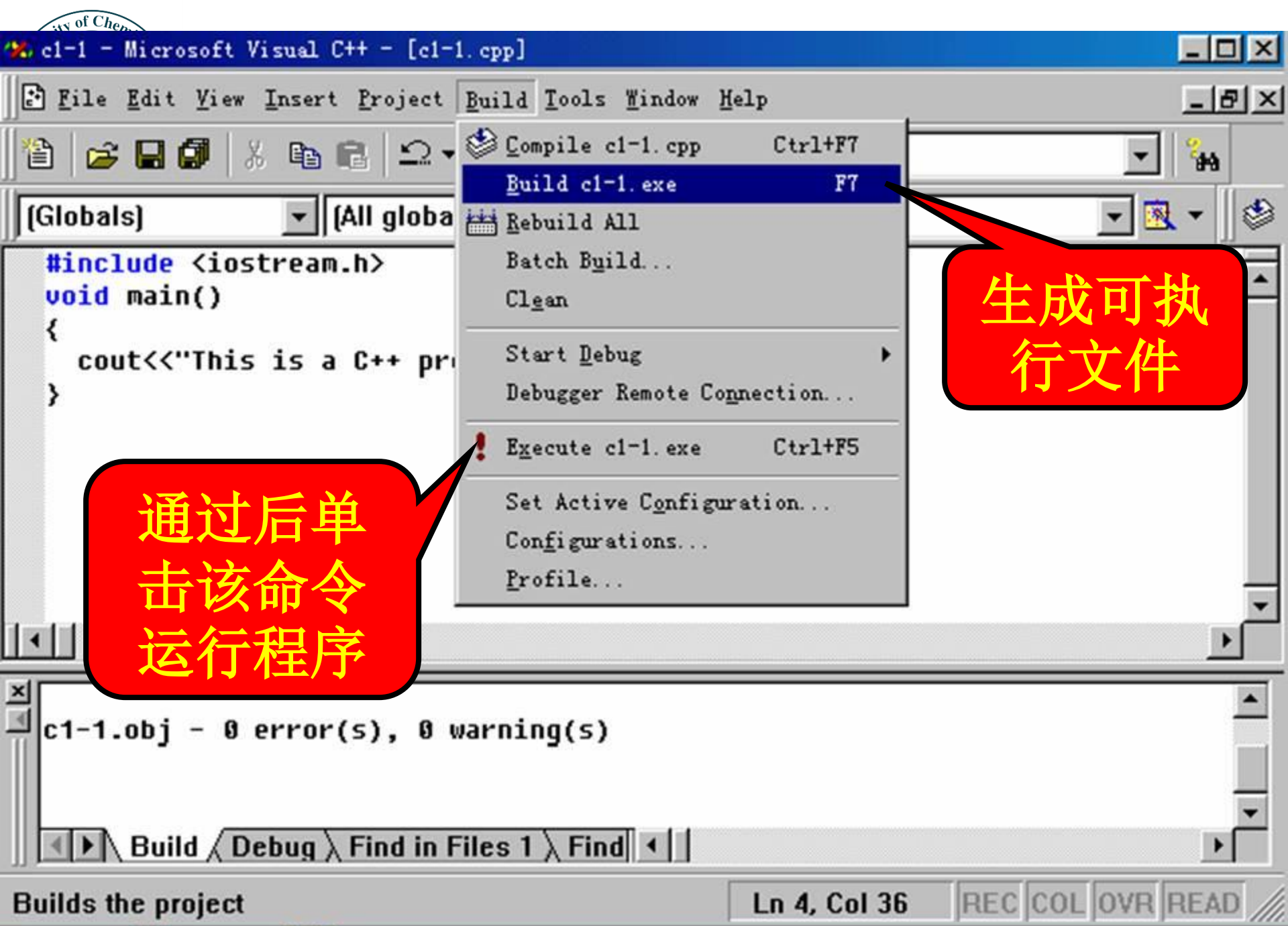
错误的原因

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)

如果编译出错，会出现提示信息，指出错误的位置及种类







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！



另一个例子

```
#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
```

```
}
```