

21 世纪

计算机应用技术系列规划教材

程序设计基础 ——C++

◎ 刘玉英 主 编 ◎

◎ 刘玉英 张怡芳 王涛伟 杨庆 编 著 ◎



人民邮电出版社
POSTS & TELECOM PRESS

21 世纪计算机应用技术系列规划教材

程序设计基础

——C++

刘玉英 主编

刘玉英 张怡芳 王涛伟 杨庆 编著

人民邮电出版社

图书在版编目 (CIP) 数据

程序设计基础: C++ / 刘玉英等编著. —北京: 人民邮电出版社, 2006.2

(21 世纪计算机应用技术系列规划教材)

ISBN 7-115-13915-6

I. 程... II. 刘... III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 142236 号

内 容 提 要

本书系统地讲述 C++ 语言的基础知识、基本语法以及编程方法, 并且结合 C++ 阐述面向对象的程序设计思想, 使读者在掌握 C++ 语言语法知识的同时, 能够解决现实生活中较简单的问题, 并用计算机语言进行描述。本书每一章中都用大量实用性较强的例题阐述基本知识点, 同时在每章的最后都提供一个有一定难度且趣味性较强的综合实例, 将本章中多个知识点有机地结合起来, 力求读者能把理论与实践紧密结合, 体会解决实际问题的过程。全书内容精练, 重点突出, 从实例出发提出概念, 突出应用, 趣味性强。

本书语言表达严谨、文字通俗易懂, 同时配有题型多样的典型习题, 适用于 C++ 高级语言程序设计的初学者, 可以作为普通高等院校中电子信息类专业程序设计基础的教材, 也适宜有兴趣学习 C++ 的非计算机专业学生作为辅助教材, 同时也适合自学。

21 世纪计算机应用技术系列规划教材

程序设计基础——C++

-
- ◆ 主 编 刘玉英
 - 编 著 刘玉英 张怡芳 王涛伟 杨 庆
 - 责任编辑 邹文波
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
 - 邮编 100061 电子函件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京通州大中印刷厂印刷
 - 新华书店总店北京发行所经销
 - ◆ 开本: 787×1092 1/16
 - 印张: 20.25
 - 字数: 484 千字
 - 印数: 1—3 000 册
 - 2006 年 2 月第 1 版
 - 2006 年 2 月北京第 1 次印刷

ISBN 7-115-13915-6/TP · 4902

定价: 26.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

C++是一种混合型的面向对象程序设计语言。它既有面向对象的特征，可以为面向对象的技术提供全面支持，又具有对传统C语言的向后兼容性，具备结构化程序设计的特征。由于C++涉及的概念多、内容广，语法比较复杂，不少初学者觉得学习难度大、入门难。

本书针对这些问题，合理安排内容，精心设计实例，引导读者一步一步地走入程序设计的知识殿堂，使读者不再感觉学习C++很累、很难，达到入门的目的。相信通过本书的讲解，您将觉得C++要比许多人想象的容易理解。

本书有四大特点。

一是详略得当，主次分明。

C++中包含的内容较多，由于篇幅的限制不可能面面俱到，必须有所舍弃。书中对于非重点或较复杂的内容略讲，如数组部分重点是一维和二维数组，三维数组只介绍概念；在结构与联合部分中，重点讲解结构，而联合的内容着重讲解与结构的区别。

二是例题生动，实用性强。

本书针对C++特点，精选重点，强化主要概念，图文并茂地讲解每个重要知识点，并配以较多容易理解的程序实例，以例题释含义、总结出规律，便于理解和应用。同时在每一章的主要内容讲解之后，充分利用前面的知识，将多个知识点有机地结合起来，设计了有一定难度并且趣味性强的综合实例，以加强对所学知识的理解和运用。第1章中，通过生活中的实例——卖报亭，渗透面向对象编程的概念；如果掌握了第2章中的内容，大家就可以设计如水果收银机那样的生活中实用的小程序。以后各章基本都有实用性较强的实例配合内容的讲解，如选择法排序和Josephus问题、贺卡设计、学生成绩管理、日期类等。通过由浅入深地讲述，生动形象的程序实例，使读者学起C++来有兴趣、有吸引力，不再感觉学习C++是很难、很枯燥的事了。本书所有的完整程序，都在C++ Builder 6.0环境下调试通过。

三是设计“思考”，启发动脑。

在典型例题之后，设计了思考题，启发引导读者进行更深入的思考，举一反三。不少读者反映，自己的编程能力差，阅读程序可以读懂，但是不会自己编写程序。作者期望通过设计思考题的方式引导读者增强编程能力。

四是配有练习，学练结合。

本书的每一章都配有适当的练习题，习题典型、多样，覆盖本章知识点。读者在学完一

章后,通过练习达到巩固强化的作用。并且在本书的附录中还提供了两套难度适当、题量适中、题型多样、覆盖面宽的模拟题以及参考答案,方便读者学完本书自测或练习。

本书从简单的面向过程(SP) C++程序,过渡到面向对象(OOP)的 C++程序,讲述了 C++语法特点、基本数据类型、运算符与表达式、数据的输入输出、控制结构、数组、函数、指针、结构、类和对象、继承与派生、多态性、模板和流类库等 C++中的基本概念和主要内容。读者可以通过学习本书掌握 C++语言程序设计的基础知识,为从入门到精通,为以后各种算法的实现,以及为进一步利用工具软件编写复杂的实用软件奠定扎实的程序设计基础。学习本书不需要有 C 语言学习经历。有 C 语言经验的读者也可以对比着学习,将能事半功倍。

本教材适用于 C++语言程序设计的初学者,可以作为普通高等院校电子信息类专业程序设计基础的教材,也适宜有兴趣学习 C++的非计算机专业学生作为辅助教材,同时也适合自学。

与本书配套的《C++实验指导与课程设计》一书中,除包含实验指导和课程设计的内容外,还提供本书习题的参考答案。两本书配合使用,学习效果更佳。

全书共 13 章,由刘玉英老师给出写作提纲和基本要求。第 1、2、6 章由张怡芳老师编写,第 3、4、7 章由王涛伟老师编写,第 8、10 章由杨庆老师编写,第 5、9、11、12、13 章由刘玉英老师编写,附录部分由刘玉英老师和杨庆老师编写、整理。最后全书由刘玉英老师改写了部分内容,统编定稿。

在“程序设计基础——C++”的教学过程中,课程组教师及相关课程的众多教师曾经举行过多次 C++课程的研讨会,作者从中受益匪浅。在此特向杨秀金等老师表示诚挚的谢意。

尽管本书作者都是多年讲授 C++课程的教师,有着比较丰富的教学经验。但是,由于受到篇幅和写作时间的限制,书中仍然可能会存在一些不足之处,恳请使用本书的教师、学生和其他读者批评指正,以便再版时修改。

我们的 E-mail 地址是: liuyy@zww.edu.cn

编 者

2005 年 12 月 26 日



第 1 章 C++简单程序设计	1
1.1 概述	1
1.1.1 C++语言的发展历史	1
1.1.2 C++程序设计的初步知识	2
1.1.3 字符集	5
1.1.4 词法记号	5
1.2 基本数据类型	6
1.2.1 基本数据类型	7
1.2.2 变量	8
1.2.3 常量	10
1.3 运算符和表达式	13
1.3.1 运算符的优先级	14
1.3.2 算术运算符及其表达式	15
1.3.3 关系运算符与逻辑运算符	15
1.3.4 增 1、减 1 运算符及其表达式	17
1.3.5 赋值运算符、复合的赋值运算符及其表达式	17
1.3.6 条件运算符	18
1.3.7 位操作运算符	18
1.3.8 其他运算符	20
1.3.9 数据类型转换	21
1.3.10 类型别名	22
1.4 面向对象设计思想及其实例	23
1.4.1 程序设计语言的发展	23
1.4.2 面向过程的程序设计 (POP) 思想	23
1.4.3 面向对象的程序设计 (OOP) 思想	24
1.4.4 面向对象思想分析实例——卖报亭	24

1.5 C++上机实践	26
1.5.1 C++程序的实现过程	26
1.5.2 C++ Builder 可视化编程环境	27
本章小结	30
习题	31
第2章 数据的输入/输出与控制结构	34
2.1 键盘输入	34
2.2 屏幕显示输出	35
2.3 字符数据的输入输出	36
2.3.1 字符数据的输入与输出	36
2.3.2 字符串的输入与输出	37
2.4 程序基本控制结构	38
2.4.1 语句的概念	38
2.4.2 算法的基本控制结构	41
2.5 选择结构	42
2.5.1 if-else 语句	42
2.5.2 switch 语句	46
2.6 循环结构	48
2.6.1 while 语句与 do-while 语句	48
2.6.2 for 循环语句	50
2.6.3 if+goto 实现循环功能	51
2.6.4 循环的嵌套	52
2.7 跳转语句	53
2.7.1 break 语句	54
2.7.2 continue 语句	54
2.7.3 break 语句与 continue 语句的比较	54
2.7.4 goto 语句	55
2.8 编程实例——水果收银机	55
本章小结	56
习题	57
第3章 数组	61
3.1 数组的基本概念	61
3.2 数组的定义与数组元素的表示法	62
3.2.1 数组的定义格式	62
3.2.2 数组元素的表示方法	62
3.3 数组的赋值	63
3.3.1 数组赋初值	63
3.3.2 数组赋值	65

3.4 字符数组.....	66
3.4.1 字符数组的定义格式.....	66
3.4.2 字符数组的赋值.....	66
3.4.3 字符数组的输入输出操作.....	67
3.4.4 字符串处理函数.....	68
3.5 编程实例——选择法排序和 Josephus 问题.....	70
3.5.1 选择法排序.....	70
3.5.2 Josephus 问题.....	71
本章小结.....	72
习题.....	72
第 4 章 函数	76
4.1 函数的定义与调用.....	76
4.1.1 函数的定义.....	76
4.1.2 函数的声明和调用.....	77
4.2 函数的调用方式和参数传递.....	79
4.2.1 函数的调用过程.....	79
4.2.2 函数的传值调用.....	80
4.2.3 函数的引用调用.....	81
4.2.4 数组作为函数参数.....	83
4.3 函数的嵌套调用和递归调用.....	85
4.3.1 函数的嵌套调用.....	85
4.3.2 函数的递归调用.....	86
4.4 带默认形参值的函数.....	90
4.5 内联函数和重载函数.....	92
4.5.1 内联函数.....	92
4.5.2 重载函数.....	93
4.6 编程实例——二进制与十进制的转换.....	94
本章小结.....	95
习题.....	95
第 5 章 程序结构	100
5.1 全局变量与局部变量.....	100
5.1.1 全局变量.....	100
5.1.2 局部变量.....	101
5.2 静态变量.....	102
5.3 存储类型.....	103
5.4 作用域与生存期.....	104
5.4.1 作用域.....	104
5.4.2 可见性.....	105

5.4.3 生存期	105
5.5 编译预处理	105
5.5.1 文件包含	106
5.5.2 宏定义	106
5.5.3 条件编译	108
5.6 多文件结构	109
本章小结	110
习题	110
第6章 指针	114
6.1 指针的概念	114
6.1.1 指针变量的声明	115
6.1.2 指针变量的初始化与引用	115
6.2 指针运算	116
6.2.1 运算符“&”和“*”	116
6.2.2 指针的运算	117
6.3 指针与数组	118
6.3.1 指针与数组的关系	118
6.3.2 通过指针引用数组元素	119
6.3.3 new 与 delete	121
6.4 字符指针	123
6.4.1 字符指针的定义	123
6.4.2 单个字符指针	124
6.4.3 指针与字符串	124
6.5 指针数组	125
6.6 指针与函数	126
6.6.1 指针用作函数参数	127
6.6.2 命令行参数 (Command-line Argument)	127
6.6.3 返回指针值的函数	128
6.6.4 指向函数的指针	129
6.7 指针与引用的区别	129
6.8 const 指针与引用	131
6.8.1 const 指针	131
6.8.2 用 const 限定引用	132
6.9 编程实例——贺卡设计	132
本章小结	134
习题	135
第7章 结构及其他	138
7.1 结构的概念	138

7.1.1 结构和结构变量的定义	138
7.1.2 结构变量成员的访问	139
7.1.3 结构变量的赋值	140
7.1.4 结构变量的运算	141
7.2 结构与数组	142
7.2.1 数组作结构成员	142
7.2.2 结构变量作数组元素	143
7.3 结构与函数	144
7.3.1 结构变量和指向结构变量的指针作函数参数	144
7.3.2 结构变量和指向结构变量的指针作函数返回值	146
7.4 其他构造数据类型	147
7.4.1 联合	147
7.4.2 枚举	149
7.5 编程实例——学生成绩管理	150
本章小结	152
习题	152
第8章 类和对象（一）	156
8.1 面向对象程序设计	156
8.1.1 面向对象的思想	156
8.1.2 面向对象程序设计的基本特点	157
8.2 从结构到类	158
8.2.1 数据抽象和结构	158
8.2.2 给结构添加函数	159
8.2.3 类与结构	160
8.3 类的声明与成员的访问	160
8.3.1 类的声明	160
8.3.2 类成员的访问控制	161
8.3.3 类的成员函数	162
8.4 成员函数的特征	163
8.4.1 内联成员函数	163
8.4.2 成员函数的重载性	164
8.4.3 具有默认参数的成员函数	165
8.5 对象	165
8.5.1 对象的定义格式	165
8.5.2 对象的成员表示	165
8.6 对象的初始化	167
8.6.1 构造函数与默认构造函数	167
8.6.2 拷贝构造函数与默认拷贝构造函数	169

8.6.3 析构函数与默认析构函数	171
8.7 友元	173
8.7.1 友元函数	174
8.7.2 友元类	176
8.8 编程实例——园林公司工程估算程序	177
本章小结	180
习题	181
第9章 类和对象（二）	186
9.1 静态类成员	186
9.1.1 静态数据成员	186
9.1.2 静态成员函数	189
9.2 常成员	191
9.2.1 常数据成员	191
9.2.2 常成员函数	192
9.3 子对象和堆对象	193
9.3.1 子对象	193
9.3.2 堆对象	194
9.4 对象数组和对象指针	195
9.4.1 对象数组	195
9.4.2 对象指针	197
9.4.3 对象引用	198
9.5 编程实例——王婆卖瓜	199
本章小结	201
习题	201
第10章 继承与派生	207
10.1 继承与派生	207
10.1.1 继承与派生的概念	207
10.1.2 派生类的声明	209
10.1.3 派生类的生成过程	210
10.2 类的继承方式	211
10.2.1 公有继承	211
10.2.2 私有继承	213
10.2.3 保护继承	215
10.3 派生类的构造函数与析构函数	217
10.3.1 派生类的构造函数	218
10.3.2 派生类的析构函数	220
10.4 多继承中的二义性问题	222
10.5 虚基类	226

10.5.1 虚基类的概念	226
10.5.2 虚基类的构造函数	228
10.6 编程实例——小型公司人员信息管理程序	229
本章小结	235
习题	236
第 11 章 多态性与虚函数	240
11.1 运算符重载	240
11.1.1 运算符重载的规则	240
11.1.2 运算符重载为成员函数	241
11.1.3 运算符重载为友元函数	242
11.1.4 其他运算符重载	243
11.2 虚函数	246
11.2.1 虚函数的定义	246
11.2.2 虚函数的使用原则	247
11.2.3 虚函数的访问	247
11.3 纯虚函数与抽象类	248
11.4 编程实例——日期类	250
本章小结	253
习题	253
第 12 章 模板与异常	257
12.1 模板的概念	257
12.2 函数模板	258
12.2.1 函数模板的声明	258
12.2.2 函数模板的应用举例	258
12.3 类模板	260
12.3.1 类模板的声明	260
12.3.2 类模板的应用	261
12.4 异常处理	262
12.4.1 异常处理机制	262
12.4.2 异常处理的实现	262
12.4.3 异常处理的应用	263
本章小结	264
习题	265
第 13 章 C++语言的输入与输出	268
13.1 输入输出标准流类	268
13.1.1 标准输出	269
13.1.2 标准输入	270

13.1.3 格式化输入与输出	272
13.1.4 输入输出运算符重载	275
13.2 文件流类	277
13.2.1 文件的打开与关闭	277
13.2.2 文本文件的读写操作	278
13.2.3 二进制文件的读写操作	280
13.3 编程实例——简单的通讯录	281
本章小结	283
习题	284
附录 1 模拟题及模拟题参考答案	287
模拟题 1	287
模拟题 2	293
模拟题 1 参考答案	299
模拟题 2 参考答案	302
附录 2 ASCII 码表	306
附录 3 常用库函数	307
参考文献	310

第 1 章

C++简单程序设计



计算机最吸引人之处是它能自动执行指定的程序。所谓程序，就是计算机解决某些特定问题所需的代码化指令序列（或称符号化语句序列），是计算机的重要组成部分。程序设计者根据预先制定的规则和功能，编写出完整的、一系列指令的有序集合，由计算机执行，实现某种特定任务，这就是计算机程序。

程序设计就是根据给定问题，设计、编制、调试程序。它包括 4 个基本步骤：①确定数据结构和算法；②编写源代码；③在计算机上调试程序；④整理，写出文档资料。

本章讲述 C++ 程序设计的初步知识，包括 C++ 语言的基本数据类型，运算符和表达式。

1.1 概述

1.1.1 C++语言的发展历史

程序设计的任务就是用计算机懂得的语言编写程序，然后交由计算机去执行，因此，程序设计语言也是程序设计必不可少的工具。C++ 语言就是一种常用的高级计算机程序设计语言。

C++ 是从 C 语言发展演变而来的。20 世纪 80 年代初，美国 AT&T 贝尔实验室设计并实现了 C++ 语言，是一个更好的 C，它引入了类的机制，因此，早期称它为“带类的 C”。此时，是在 C 语言的基础上加进类及派生类，特点是区分公有成员、私有成员，有了类的构造函数和析构函数、友元、内联函数等内容，同时还包括运算符重载。从带类的 C 到 C++，又添加了虚函数、函数重载、引用和常量等内容。1983 年正式取名为 C++，并于 1985 年正式推出它的 1.0 版本。C++ 语言的标准化工作从 1989 年开始，于 1994 年制定了 ANSI C++ 标准草案。以后又经过不断完善，成为目前的 C++。实际上，从 1989 年 C++ 的 2.0 版到 1993 年 C++ 的 3.0 版是 C++ 的发展与完善阶段，这时已经包含了面向对象的所有概念，有类的保护成员、多重继承、对象的初始化、抽象类、静态成员、const 成员和模板等。C++ 语言的发展历史如图 1.1 所示。

在 C 语言的基础上发展起来的 C++ 引入了面向对象的机制，成为优秀的面向对象程序设计语言，但它也支持面向过程的程序设计。它的面向对象的程序设计方法，很好地解决了日

益复杂的程序设计问题，在 20 世纪 70 年代末，这种面向对象的设计方法较好地解决了“软件危机”问题。

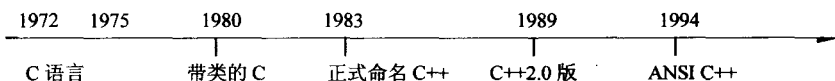


图 1.1 C++语言的发展历史

1.1.2 C++程序设计的初步知识

为了让读者对编程语言有初步的认识，下面通过几个简单的例子，介绍 C++程序设计语言的特点。

1. 一个最简单的 C++程序

【例 1.1】 最简单的 C++语言程序，在屏幕上显示英文句子“Hello, C++!”。

```
#include <iostream.h>
void main ()
{
    cout<<" Hello, C++ !" <<endl ;
}
```

这是一个最简单的 C++语言程序，它的主要功能是在屏幕上输出英文句子：

Hello, C++ !

在这个程序中，虽然它只有一条语句，但它是一个完整的程序，经过编译、连接后可以运行。其中，“iostream.h”是标准输入输出(I/O)流类库提供的文件，该头文件中包含对“cout”和运算符“<<”的定义。程序中“cout<<”的作用就是在屏幕上显示字符串“Hello, C++!”，而“endl”（读为英文：end line）的意义为换行。

2. 其他简单的 C++程序

面向对象的 C++语言是从过程化的 C 语言发展而来的，因此它既可以实现面向过程的程序设计，也可以实现面向对象的程序设计。下面的程序中既有面向过程的编程，也有面向对象的编程。在这里，对这些程序的要求只是“认识”它们，借此了解 C++的程序结构，并不要求掌握它们。

【例 1.2】 根据输入的半径长度求圆的面积。

```
#include <iostream.h>
void main()
{
    double s,r;
    cout<<"Input the radium: r= "; //提示输入半径
    cin>>r;
    s=3.14159*r*r;
    cout<<"r= " <<r<<endl; //输出半径
    cout<<"the area is " <<s<<endl; //输出面积
}
```


运行该程序时，屏幕会显示提示：

Input the radium: r=

5 ↵ (表示从键盘上输入 5 后回车) 将显示：

r=5

the area is 78.5397

请注意，在本书中，一律用“↵”表示回车（按 Enter 键）。

【例 1.3】 调用自定义函数求两个数之和。

```
#include <iostream.h>
int add2 ( int x , int y );           //声明自定义函数 add2()
void main ()
{   int  a , b , c ;                  //定义整型变量 a , b , c
    cin>>a>>b;                       //从键盘上输入变量 a , b 的值
    c = add2(a , b ) ;                //调用自定义函数 add2
    cout<<" 它们的和为" <<c<<endl ;
}
int add2 ( int x , int y )            //对 add2 函数的定义
{   int  z ;                          //定义整型变量 z
    z=x+y;                           //求变量 x 加 y 的和，并将和赋予变量 z
    return  z ;                      //返回变量 z 的值
}
```

本程序运行时需要输入两个数，屏幕将等待输入两个数据。当输入

3 4 ↵ 将显示：

它们的和为 7

【例 1.4】 基于对象的程序设计：定义一个 dog 类，并使用这个类。

```
#include <iostream.h>
class dog                             //定义一个 dog 类
{   private:
    char *name;
    int age;
    char sex;
    public:
    dog (char *n, int a=0, char s='m');
    void speak();
};
dog::dog (char *n, int a, char s)     //定义 dog 类的构造函数
{   name=n; age=a; sex=s; }
void dog::speak()                    //定义成员函数
{   cout<<"My name is "<<name<<endl;
    cout<<"I can speak: woof woof woof !"<<endl;
```

```

}
void main()
{   dog mypet( "lili" , 2, 'f');           //mypet 是 dog 类的一个对象（实例）
    mypet.speak();
}

```

程序的执行结果是：

My name is lili

I can speak: woof woof woof!

注意：以上每个程序要运行时，必须先经过源程序编辑，用不同的文件名分别储存在计算机中，然后通过语言编译器生成目标文件，再经过连接生成可执行文件后才能执行。

用高级语言的语句组成的序列通常称为源程序。所谓源程序，即程序设计者按照一定的语言的规范编写的原始数据代码，也称源文件。一般源程序都是 ASCII 文本，故也称为源代码。同时，编译 C++ 语言程序一般需要一个集成环境（IDE）。例如，C++ 程序可以在 Visual C++ 或 C++ Builder 所提供的集成环境中运行。

3. C++ 程序结构

每个 C++ 程序一般由程序主体、编译预处理和注释三部分组成。

（1）程序主体：函数定义是 C++ 程序的主体部分。程序的作用基本上由函数来完成。

（2）编译预处理：每个以符号“#”开头的行，称为编译预处理行，是 C++ 提供的一种模块化工具。

（3）注释：注释的作用是用来给程序设计者一种提示或记号，注释部分内容不参加程序的编译和执行。C++ 提供以下两种注释形式。

- 一对“/*”和“*/”之间的内容为注释部分，它可以作用于多行，是从 C 语言中继承来的注释形式。

- “//”符号之后的该行内容为注释部分，仅本行有效，是 C++ 语言特有的。

4. C++ 程序结构特点

C++ 语言全面兼容 C 语言，它保持了 C 语言的简洁、高效和接近汇编语言等特点，又对 C 语言的类型系统进行了改革和扩充，它不是一个纯正的面向对象的语言，适用性广。C++ 程序结构特点如下：

- C++ 程序是一个函数串，一个 C++ 程序可以包括任意多个不同名的函数，但是有且仅有一个主函数 main()，程序执行时，从主函数 main() 开始。

- 函数可以自定义，“{”和“}”分别是函数体开始和结束的标志。也可以调用 C++ 程序已有的库函数等。

- 语句是组成程序的基本单元，语句都以英文分号（;）结束，每一条语句代表一种操作。

- 书写格式自由，可以在一行的任意位置书写；一行可以写一个语句，也可以写多个语句。

- 注释用“//”或一对“/*”“*/”的形式，编译时系统对注释部分不作处理。

- 一般都需要包含由系统提供的头文件，如 #include <iostream.h>、#include <conio.h> 等。