

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

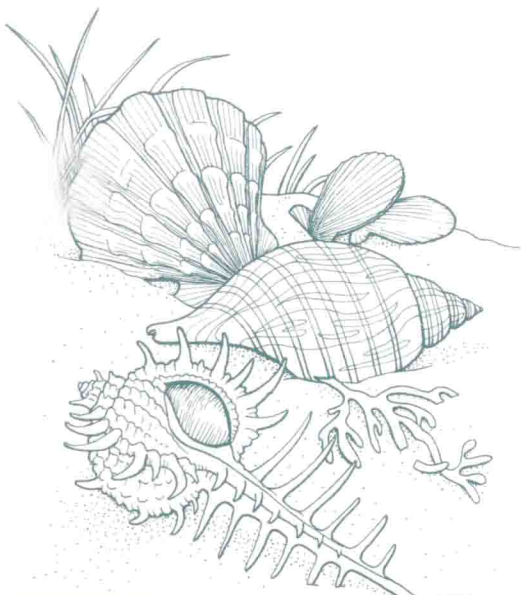
C语言程序设计 实训教程

Training Course of C Programming Language

李振立 张慧萍 编著

张颖江 主审

- 优化知识体系-知识点全面、案例丰富，以编程方法为主线，贯穿全文。
- 增强实用性-采用案例驱动阐述方法，以典型案例引导，由浅入深展开教学。
- 创新编写思想-以内存映射图解程序语句的操作过程，采用图解法激发逻辑思维与形象思维能力。



高校系列

 人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

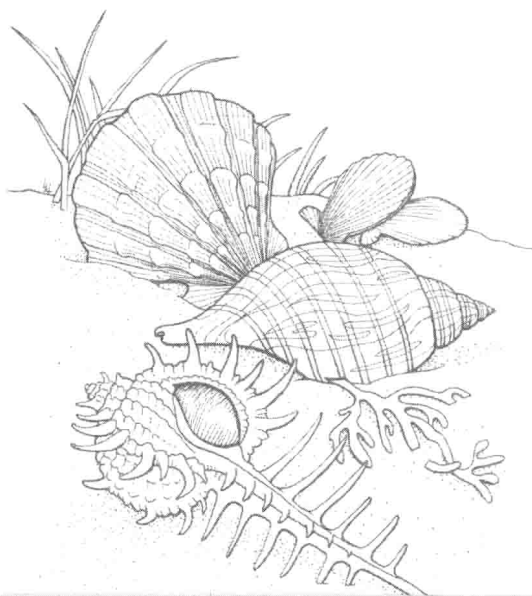
21st Century University Planned Textbooks of Computer Science

C语言程序设计 实训教程

Training Course of C Programming Language

李振立 张慧萍 编著

张颖江 主审



高校系列

人民邮电出版社

图书在版编目 (C I P) 数据

C语言程序设计实训教程 / 李振立, 张慧萍编著. —
北京: 人民邮电出版社, 2014. 10
21世纪高等学校计算机规划教材
ISBN 978-7-115-36594-1

I. ①C… II. ①李… ②张… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第194070号

内 容 提 要

本书是与《C语言程序设计》相配套的实验教程, 全书共10章, 每章由学习指导和实训两部分构成, 学习指导包括预习指导、学习指导、典型案例分析、解题方法、分析方法、习题解答、机考模拟题等内容。实训包括实验要求、编程方法、程序调试方法、实验报告等内容。本书从预习、学习、复习、练习、上机、调试和测试各个教学环节为学生提供案例和学习参考内容, 训练学生的计算思维能力、分析能力和编程能力。

本书由长期工作在教学一线的教师编写, 全书各知识单元编排层次清楚、条理分明、结构合理而严谨, 案例丰富, 详略度把握得体, 与《C语言程序设计》配套形成理想的C语言程序设计的教材。本书既可作为各类高等院校本、专科非计算机专业的C语言程序设计的教材, 也可以作为独立学院、高职高专、网络学院的教材。

◆ 编 著 李振立 张慧萍

主 审 张颖江

责任编辑 邹文波

执行编辑 吴 婷

责任印制 彭志环 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

大厂聚鑫印刷有限责任公司印刷

◆ 开本: 787×1092 1/16

印张: 18.75

2014年10月第1版

字数: 494千字

2014年10月河北第1次印刷

定价: 39.80元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前言

本书是与《C 语言程序设计》相配套的实验教程。全书各知识单元编排与教材相同，内容互补，结构合理而严谨，案例丰富，具有知识性与趣味性，是理想的 C 语言程序设计的配套教材。

本书分为 10 章，分别为 C 语言概述指导与实训、数据类型与表达式指导与实训、顺序结构程序设计指导与实训、选择结构程序设计指导与实训、循环结构程序设计指导与实训、数组指导与实训、函数指导与实训、指针指导与实训、用户定义数据类型指导与实训、文件指导与实训。

本书通过大量的教学案例，为学生提供较为全面的程序设计思想、程序设计分析方法和解题思路；着重介绍以内存分析为主的图解法，各个章节都可以使用图解分析方法解析程序的存储与操作过程，将源程序中的每条语句解析为对内存数据的操作映射；以集成开发环境 Visual C++ 6.0 为开发工具，介绍了 VC 的编辑、编译、运行、排错、调试等使用方法，帮助学生熟悉 C 语言编译器，快速掌握编辑、编译、运行 C 程序的方法，学会使用集成开发环境开发简单应用软件；从预习、学习、复习、练习、上机、调试和测试各个教学环节为学生提供案例和学习参考内容，训练学生计算思维能力、分析判断能力和应用编程能力。

C 语言的练习题目，如全国计算机等级考试 C 语言试题、各种考试的 C 语言试题在网上数不胜数，因此本书将题目考点进行归类总结，对试题进行分类、解析、剖析经典案例，让学生学会分析方法和编程方法，举一反三、融会贯通，品味到解题、编程的乐趣。

本书既可作为各类高等院校本、专科非计算机专业的 C 语言程序设计的教材，也可以作为独立学院、高职高专、网络学院的教材。对于社会计算机学习者，尤其是具有一定计算机基础而又欲获得提高的广大计算机爱好者，本书无疑是一本极好的自学教材。

本书由长期工作在教学一线的教师编写，由张颖江主审，李振立、张慧萍主编，其他编委还有程学先、楚维善、李光明、顾梦霞、陈小娟、贺红艳、吴佩、罗宏芳、李珺、陈锦、周雪琴、王世畅。在本书的策划、编写、出版过程中，湖北工业大学工程技术学院的各级领导给予了大力支持，在此深表谢意。

编者

2014 年 7 月 8 日

目 录

第 1 章 C 语言概述指导与实训 1

1.1 教材的预习及学习指导 1

1.1.1 教材预习指导 1

1.1.2 教材学习指导 1

1.2 分析方法 7

1.2.1 案例分析 7

1.2.2 编程方法 9

1.3 练习题与解答 12

1.3.1 教材练习题解答 12

1.3.2 网络考试样卷与考生须知 16

1.4 C 语言基本操作实验 27

1.4.1 集成开发环境 VC++ 27

1.4.2 实验报告 31

第 2 章 数据类型与表达式指导与实训 36

2.1 教材的预习及学习指导 36

2.1.1 教材预习指导 36

2.1.2 教材学习指导 36

2.1.3 补充教材——数制与编码 41

2.2 分析方法 46

2.2.1 案例分析 46

2.2.2 编程方法 51

2.3 练习题与解答 54

2.3.1 教材练习题解答 54

2.3.2 机考练习题 58

2.4 数据类型与表达式实验 63

2.4.1 程序调试方法 63

2.4.2 实验预习与实验报告 65

第 3 章 顺序结构程序设计指导与实训 69

3.1 教材的预习及学习指导 69

3.1.1 教材预习指导 69

3.1.2 教材学习指导 69

3.2 分析方法 72

3.2.1 案例分析 72

3.2.2 算法分析 74

3.3 练习题与解答 77

3.3.1 教材练习题解答 77

3.3.2 机考练习题 81

3.4 顺序结构程序设计实验 87

3.4.1 顺序结构程序设计 87

3.4.2 实验报告 90

第 4 章 选择结构程序设计指导与实训 93

4.1 教材的预习及学习指导 93

4.1.1 教材预习指导 93

4.1.2 教材学习指导 93

4.2 分析方法 96

4.2.1 案例分析 96

4.2.2 算法分析 99

4.3 练习题与解答 102

4.3.1 教材练习题解答 102

4.3.2 机考练习题 107

4.4 选择结构程序设计实验 114

4.4.1 选择结构程序设计 114

4.4.2 预习作业与实验报告 117

第 5 章 循环结构程序设计指导与实训 121

5.1 教材的预习及学习指导 121

5.1.1 教材预习指导 121

5.1.2 教材学习指导 122

5.2 分析方法 123

5.2.1 算法分析 123

5.2.2 案例分析 126

5.3 习题与解答 130

5.3.1 练习题 130

5.3.2 机考练习题 135

5.4 循环结构程序设计实验 142

5.4.1 循环结构程序设计	142	8.2.2 案例分析	218
5.4.2 预习作业与实验报告	146	8.3 习题与解答	221
第 6 章 数组指导与实训	150	8.3.1 练习题	221
6.1 教材的预习及学习指导	150	8.3.2 机考练习题	226
6.1.1 教材预习指导	150	8.4 指针实验	233
6.1.2 教材学习指导	150	8.4.1 编程方法	233
6.2 分析方法	154	8.4.2 实验报告	236
6.2.1 算法分析	154	第 9 章 用户定义数据类型指导与实训	241
6.2.2 案例分析	157	9.1 教材的预习及学习指导	241
6.3 习题与解答	162	9.1.1 教材预习指导	241
6.3.1 练习题	162	9.1.2 教材学习指导	241
6.3.2 机考练习题	166	9.2 数据类型分析	244
6.4 数组实验	173	9.2.1 结构体数据类型	244
6.4.1 数组编程方法	173	9.2.2 共用体数据类型	245
6.4.2 实验与实验报告	176	9.2.3 自定义类型	246
第 7 章 函数指导与实训	180	9.2.4 枚举类型	246
7.1 教材的预习及学习指导	180	9.2.5 链表	247
7.1.1 教材预习指导	180	9.3 习题与解答	247
7.1.2 教材学习指导	180	9.3.1 练习题	247
7.2 分析方法	184	9.3.2 机考练习题	252
7.2.1 数据传递分析	184	9.4 用户定义数据类型实验	260
7.2.2 案例分析	187	9.4.1 编程方法	260
7.3 习题与解答	190	9.4.2 实验报告	265
7.3.1 教材练习题	190	第 10 章 文件指导与实训	269
7.3.2 机考练习题	195	10.1 教材的预习及学习指导	269
7.4 函数实验	203	10.1.1 教材预习指导	269
7.4.1 函数编程方法	203	10.1.2 教材学习指导	269
7.4.2 实验报告	208	10.2 习题与解答	273
第 8 章 指针指导与实训	212	10.2.1 练习题	273
8.1 教材的预习及学习指导	212	10.2.2 机考练习题	278
8.1.1 教材预习指导	212	10.3 文件实验	286
8.1.2 教材学习指导	212	10.3.1 文件编程方法	286
8.2 分析方法	216	10.3.2 实验预习与实验报告	290
8.2.1 指针分析	216		

第 1 章

C 语言概述指导与实训

1.1 教材的预习及学习指导

1.1.1 教材预习指导

本课程教材《C 语言程序设计》从程序构成入手，围绕 C 语言的典型程序，由浅入深系统地介绍编程方法。第 1 章主要介绍 C 语言的程序构成、程序样板、程序逻辑顺序、程序简单的输入/输出、程序风格、程序模块和程序的组成元素；介绍 C 语言词法中的单词，简单介绍 C 语言常用的编译器如 VC++，DEV-C++等。

本章分为 4 节，第 2 节 C 语言的程序构成是预习的重点。

第 1 节 C 语言的发展史，预习时只需浏览 C 语言的起源、C 语言的特点、C 语言的集成开发环境等内容。第 2 节 C 语言程序构成是预习的重点，应从 C 语言程序的基本构成入手，了解 C 语言函数模块一般形式。通过读程序，了解什么是编译预处理命令，什么是函数的定义。函数中声明部分用于声明变量或函数，C 语言中变量或函数必须要满足先声明（定义）后使用的规则，了解函数执行语句中常用的基本语句。注意 C 语言程序中输入、处理和输出三者的逻辑顺序。学习 C 语言的书写风格。第 3 节 C 语言的单词，单词包括分隔符、注释符、关键字、标识符、常量、运算符等。预习的重点包括 C 语言基本字符集、关键字、标识符、常量与常量的类型、运算符的使用方法，掌握 C 语言的词法构成。第 4 节 DEV-C++集成开发环境，预习的重点是 DEV-C++的使用方法，新建源文件，编辑源文件，编译和调试文件，运行文件。

1.1.2 教材学习指导

1. C 语言基本概念

- C 语言集高级语言和低级语言的优点于一身，适合作为系统描述语言，用于编写大型的操作系统、编译系统、应用软件。
- C 语言属于面向过程的程序设计语言，采用结构化、模块化的方法设计源程序。
- 面向过程的程序 = 算法 + 数据结构。
- ISO：国际标准化组织。
- ANSI：美国国家标准协会。

- GNU：是一个自由软件工程项目，由自由软件社团开发和维护。
- “K&R”是以名著《The C Programming Language》的作者命名的早期 C 语言的事实标准，称为经典 C 语言。
- C 语言的美国国家标准 ANSI C 是 1983 年美国国家标准协会（ANSI）对 C 语言进行扩充和规范制定的标准。
- C89 是指 1989 年 ISO/IEC 提出的国际标准草案，1990 年公布的 C 语言正式标准称为 C89，有时也称为 C90。
- C99 是指 1999 年 12 月 16 日 ISO/IEC 推出的 C 语言标准：ISO/IEC 9899:1999（Programming languages-C）。
- C 语言具有语言简洁、紧凑，使用灵活、方便，运算符丰富，表达能力强，数据结构丰富，结构化程度高、生成目标代码质量高，程序执行效率高、可移植性好等诸多特点。
- C 语言集成开发环境 IDE 都是由编辑器、编译器、连接器集合而成的。常用的 C 与 C++ 集成开发环境有 Borland 公司的 Turbo C、Borland C++、C++ Builder；微软的 Microsoft C、Visual C++、Visual studio.NET、VC2005、VC2008、VC2010、VC2012；开源社区 GNU 的 C/C++ 集成开发环境有 GCC、DEV-C++、CodeBlocks、Kdevelop、Anjuta Devstudio、Visual-MinGW、MinGW Developer Studio、QT、Tiny C compiler、Eclipse+CDT、Netbeans C++ 等软件。
- C 程序的基本构成为编译预处理命令、参数说明、函数说明、主函数、函数。其中，主函数由函数头部、函数体组成，函数头部包括函数类型、函数名、参数说明；函数体由声明语句、执行语句等组成，声明是对定义的变量或函数使用之前的说明，声明也是一种定义。
- 编译预处理命令有文件包含、宏定义和条件编译 3 类命令。
- C 语言以函数作为语言单位，主程序和程序模块都是函数。
- 函数定义的参数为虚参，函数调用的参数为实参，参数按位置虚实对应进行传递。
- 变量、函数应该满足先声明后使用的规则，如果主函数在前，函数的定义在后，需要用函数原型声明函数，说明函数的类型，参数类型和个数。
- C 语句用分号“;”作结束符，编译预处理命令结束没有分号“;”。
- 注释分为块注释和行注释，块注释用一对符号“/* */”作程序中注释的定界符，表示“/*”和“*/”之间的内容是注释内容；用“//”引导行注释，表示从“//”开始到行尾为注释内容。
- C 程序的逻辑顺序按数据声明、数据输入、数据处理、数据输出的次序排列。
- 声明变量和函数，确定数据的类型和取值范围，声明变量时可以对变量初始化。数据输入包括函数参数传递的数据，变量初始化时输入的数据，输入函数和赋值语句输入的数据等。数据处理是根据解题的算法编制的程序语句，数据输出是用输出函数输出指定的数据。
- 程序样板指 C 源程序中存在一些使用方法相对固定的成分，包括编译预处理命令、主函数头部、声明语句、执行语句和返回语句等语法成分，包括一些相对固定的算法。将源程序的这些固定成分作为程序样板，对其中的某些语法成分稍作修改，可以快速地编制出解决实际问题的源程序。
- C 语言的风格指 C 程序的风范格局，是程序员对程序书写形式一贯性的体现。
- C 语言符号体系的基础是基本字符集，基本字符集是编写源程序时准用字符的集合，C 语言编译程序能够识别集合中的字符。准用字符包括大写字母、小写字母、数字、空白符、图形符号等。转义符号是由“\”开头，后跟指定的字符表示转换成另外意义的符号。
- 单词是基本字符集中的若干字符组合成的一个具有独立意义的最小词法单位，是一组形

式化的数据符号。单词包括分隔符、注释符、关键字、标识符、常量、运算符六类。

- 词法分析负责从构成源程序的基本字符集中识别和分离出单词,为语法分析提供单词类别和单词自身值的信息。词法分析对源程序进行行编辑,删除源程序中的注释、空白符以及对语法分析无关的信息。区别单词是关键字还是标识符,为语法分析做准备。

- 分隔定界符指具有分隔作用和定界作用的字符,分隔和界定程序的正文、语句或单词等语法成分。

- 空白符包括空格符、水平制表符(HT)、垂直制表符(VT)、回车符(CR)、换行符(LF)、换页符(FF)等,空白符用于语句行之间的分割。

- 常用的标点符号包括逗号、分号、冒号、小括号()、花括号{}等。

- 关键字是由编译程序预先定义具有固定含义的单词,关键字有特定含义和专门用途,用户不能用关键字作为常量、变量、类型或函数的名字。

- 标识符是定义符号常量、标号、变量、类型、函数、对象的定义符。

- 标识符的命名规则是以英文字母或下划线开头的英文大小字母、数字和下划线符组成的序列,标识符中的字母区分大小写,标识符的长度为1~31字节,C99标准规定标识符的长度为1~63字节。标识符的中间不能有分割符,不能使用C的关键字作为标识符。

- 常量指在程序运行和处理的过程中其值始终不能被改变的量。

- 常量的基本类型包括整型常量、字符型常量、字符串常量、浮点型常量4种基本类型。

- 整型常量是由字符和数字组成的序列,用以表示整数,包括正整数、负整数和零。

- C语言整型常量包括十进制、八进制和十六进制3种数制表示方法。

- 整型常数的后缀有L(或l)作后缀表示长整数,U(或u)作后缀表示无符号数。整型常量后缀可以是U和L(u或l)的组合,表示unsigned long类型的常量。

- 浮点型常量指带有小数的十进制数。

- 浮点型常量有两种表示形式。一种是带小数的十进制表示形式,它由整数部分和小数点后的尾数部分组成。另一种方法是指数形式,由带小数的十进制数后加e(E)及整数的指数部分表示,e(E)指数部分表示10的整数次方,如1.234e2表示 1.234×10^2 。

- 浮点型常量分为单精度(float)、双精度(double)和长双精度(long double)3类,浮点型常量在默认情况下为double型,若要表示float型常量,则在实数后加f(F);表示long double型则在实数后加l(L)。

- C字符常量是用单引号括起来的一个基本字符。

- 字符常量有两种表示形式,一种是准用字符表示形式,用单引号括起准用字符,如'a';另一种是转义符号表示形式,用单引号括起转义符号,如:'\n','\t','\b'等。

- 字符串常量是用一对双引号括起来的字符序列。例如,"Hello!","a"等。字符序列中的字符包括空格符、准用字符、转义字符和扩展的ASCII码字符等。

- 符号常量按照先定义后使用的原则,先用"#define 标识符 字符串"进行宏定义,然后在程序中使用符号常量。例如"#define PI 3.1415926"。

- C语言运算符分为单目运算符、双目运算符、三目运算符。单目运算符指只对一个操作数进行操作的运算符;双目运算符是对两个操作数进行操作的运算符;三目运算符是对三个操作数进行操作的运算符。需要操作数最多的运算符是三目运算符即条件运算符,条件运算符格式为“表达式1?表达式2:表达式3”。

● 运算符分为十五种优先级和两类结合性，优先级指运算符的优先次序，结合性指当一个运算量两边的运算符级别相同时指定的结合方向。

2. C 语言语法与词法

(1) C 语言的函数模块一般形式见表 1-1。

表 1-1

C 语言函数模块

函数模块格式	源程序	函数样板
编译预处理命令 函数类型 函数名 (函数参数) { 声明语句 执行语句 }	<pre>#include <stdio.h> int main() { int iA=6, iB=8, iC; iC = iA+iB; printf("iC=%d\n",iC); return 0; }</pre>	<pre>#include <stdio.h> int main() { return 0; }</pre>
函数类型 函数名 (参数表列) { 声明语句 执行语句 [return 语句] }	<pre>int iMax(int iX , int iY) { int iZ; if(iX>iY) iZ = iX; else iZ = iY; return (iZ); }</pre>	<pre>int iMax(int iX , int iY) { return (); }</pre>

主函数样板由编译预处理命令，主函数头部、函数体开始与函数体结束等部分组成。主函数的类型为整型，主函数的返回值为 0，代表正常结束；主函数要受上一级程序调用，上一级程序包括操作系统、IDE、调试器和各种自动测评系统，接收到正常结束的返回值 0 后，确认源程序正常结束。在各种信息与算法大赛中，最好用主函数为整型的函数样板。

(2) 编译预处理命令格式，见表 1-2。

表 1-2

编译预处理命令

命令格式	命令实例	功能
#include <filename.h>	例: #include <stdio.h>	从标准库目录开始搜索
#include "filename.h"	例: #include "stdlib.h"	从用户工作目录开始搜索
#define 标识符 字符串	例: #define PI 3.1415926	宏定义，定义符号常量 PI

(3) 常用程序语句，见表 1-3。

表 1-3

常用程序语句

命令类型	语句格式	语句
输入	scanf ("输入格式", 输入项地址表列);	scanf ("%d,%d",&iA, &iB);
赋值	变量 = 表达式 ;	iC = iA+iB;
输出	printf ("输出格式", 输出表列);	printf("iC=%d\n",iC);

(4) C 语言基本字符集，见表 1-4。

表 1-4 基本字符集

类型	基本字符
大写字母	A B C D E F G H I J … X Y Z
小写字母	a b c d e f g h i j … x y z
数字	0 1 2 3 4 5 6 7 8 9
空白符	空格符、水平制表符 (HT)、垂直制表符 (VT)、回车符 (CR)、换行符 (LF)、换页符 (FF)
转义符号	\n 换行 (LF) \t 水平制表 (HT) \a 响铃 (BEL) \b 退格 (BS) \f 换页 (FF) \r 回车 (CR) \v 垂直制表 (VT) \\ 反斜杠 \' 单引号 \" 双引号 \? 问号 \0 空字符 (NULL) \ddd 三位八进制数 \xhh 二位十六进制数
图形符号	~ ! # % & ? ^ * = + - _ / \ “ ‘ () [] { } < > , . : ; ?

(5) 分隔符，见表 1-5。

表 1-5 分隔符

类型	字符
空白符	空格符、水平制表符 (HT)、垂直制表符 (VT)、回车符 (CR)、换行符 (LF)、换页符 (FF)
标点符号	逗号(,)、分号(;)、冒号(:)、花括号{ }

(6) 注释符，见表 1-6。

表 1-6 注释符

注释名	符号	举例	说明
块注释	/* … */	/* 注释内容*/	所有 C 标准
行注释	// …	// 注释内容直到行尾	C99 标准

(7) 关键字，见表 1-7。

表 1-7 关键字

关键字类型	保留字
C 关键字 (基本人集) 32 个	auto、break、case、char、const、continue、default、do、double、else、enum、extem、float、for、goto、if、int、long、register、return、short、signed、sizeof、static、struct、switch、typedef、union、unsigned、void、volatile、while
Turbo C 扩展 7 个关键字	asm、cdecl、far、huge、interrupt、near、pascal
VC++扩展 34 个关键字	asm、bad_cast、bad_typeid、bool、catch、class、const_cast、delete、dynamic_cast、except、explicit、false、finally、friend、inline、mutable、namespace、new、operator、private、protected、public、reinterpret_cast、static_cast、template、this、throw、true、try、type_info、typeid、typename、virtual、wchar_t
C99 新增 5 个关键字	restrict、inline、_Complex、_Imaginary、_Bool

(8) 标识符，见表 1-8。

表 1-8

标识符

对象	定义格式	定义举例	标识符
符号常量	#define 标识符 字符串	#define PI 3.1415926	PI
标号	标号名:	Lable:	Lable
变量与数组	类型名 变量名, 数组	int iK, iA[3]={1,2,3}	iK、iA
函数	[类型说明符] 函数名 ([形参表])	int iMax(int iA, int iB)	iMax

(9) 常量, 见表 1-9。

表 1-9

常量

对象	常量格式	常量举例	后缀
整型常量	ddd、Oddd、0xdd	-123、0384、0x4A	73L 长整型、65U 无符号数
字符型常量	'基本字符'、'转义符'	'A'、'b'、'\n'、'\t'	
字符串常量	"字符序列"	"Hello!"	文末符\0 表示字符串结束, 不显示
浮点型常量	±ddd.dd、±ddd.dde±dd	-123.45、1.2e4	-3.8f 单精度; 6.9L 长双精度

(10) 运算符, 见表 1-10。

表 1-10

运算符

优先级	名称	运算符	举例	说明	结合性
最高 1	小括号	()	(a + b) * c	先(a + b)再* c	左到右
1	数组下标	[]	array[4]	array[0]…array[3]	左到右
1	函数调用	函数名()	x=func(a, b)	函数返回值赋给 x	左到右
1	成员运算符	. (结构或联合成员)	Stu.num	表示成员变量	左到右
1	指向运算符	-> (结构或联合成员)	p->num	p 指向的成员变量	左到右
1	自增 (后缀)	(变量)++	a ++	先执行表达式 a 再加 1	左到右
1	自减 (后缀)	(变量)--	b --	先执行表达式 b 再减 1	左到右
2	逻辑非	!	a=2; !a	a 非 0 为真, !a 为 0 假	右到左
2	按位求反	~	~a	对所有位取反 FFFE	右到左
2	一元正号	+	+5	取正值	右到左
2	一元负号	-	-8	取负值	右到左
2	自增 (前缀)	++(变量)	++ a	a 先加 1 再执行表达式	右到左
2	自减 (前缀)	--(变量)	--b	b 先减 1 再执行表达式	右到左
2	取地址	&变量	&a	取变量 a 的地址	右到左
2	取内容值	*指针	*p	取 p 指向的内容	右到左
2	强制类型转换	(类型)	(float)a	将 a 强制转换成单精度	右到左
2	长度运算符	sizeof(a)	sizeof(a)	求变量 a 的所占字节数	右到左
3	乘、除、整除、求余	*, /、/, %	a*b、a/2.0、a/b、a%b	整数/整数为整除	左到右
4	加、减	+, -	a+b、a-b	求和、求差值	左到右
5	位左移、位右移	<<、>>	a<<2、b>>3	a 左移 2 位, b 右移 3 位	左到右

续表

优先级	名称	运算符	举例	说明	结合性
6	关系	<, <=, >, >=	b<3、b<=4、b<2、b>=4	除=、!=外的其他关系	左到右
7	等于、不等于	==, !=	a==4、a!=4	等于、不等于关系	左到右
8	按位与	&	a & b	所有位做与运算	左到右
9	按位异或	^	a ^ b	所有位做异或运算	左到右
10	按位或		a b	所有位做或运算	左到右
11	逻辑与	&&	a && b	非 0 为真 0 为假与运算	左到右
12	逻辑或		a b	非 0 为真 0 为假或运算	左到右
13	条件（三目运算符）	?:	E1 ? E2 : E3 a>b?a:b	E1 为 1 取 E2 的值 E1 为 0 取 E3 的值	右到左
14	赋值	=, +=, -=, *=, /=, %=, &=, =, ^=, ~=, <<=, >>=	a=b+3 a+=4	赋值和复合赋值运算	右到左
15	逗号	,	E1, E2, ..., En	表达式的值为 En 的值	左到右

1.2 分析方法

1.2.1 案例分析

1. 用转义符表示的字符

例 1.2.1 字符串中包含转义字符，分析简单 C 语言源程序的输出结果（文件名 sy1_1.c）。

```
#include <stdio.h> // 包含标准输入输出头文件；
main() // 经典 c 的主函数头部；
{ // 函数体开始；
    printf("He\154\x6C\157! \x6Fk!\n"); // 调用标准输出函数；
} // 函数体结束。
```

解：在输出的字符串中，\154 是八进制数表示字符"l"，\x6C 是十六进制数表示字符"l"，\157 是八进制数表示字符"o"，\x6F 是十六进制数表示字符"o"。源程序编译、运行后，输出结果为：

```
Hello! ok!
```

2. 用转义符表示的控制符

例 1.2.2 输出字符串中包括用转义符表示的控制符，分析下面 C 语言源程序的输出结果（文件名 sy1_2.c）。

```
#include <stdio.h> // 包含标准输入输出头文件；
main() // 经典 c 的主函数头部；
{ // 函数体开始；
```

```
printf("字符串中的转义符\t 表示跳过一个制表位;\n 表示换行; 执行到\0 结束, 不输出后面的字符内
容。"); // 调用标准输出函数;
} // 函数体结束。
```

解: 在输出的字符串中, \t 表示跳过一个制表位; \n 表示换行; \0 表示结束字符串的显示。
源程序编译、运行后, 输出结果为:

字符串中的转义符 表示跳过一个制表位;
表示换行; 执行到

3. 用赋值语句计算和处理数据

例 1.2.3 已知电路参数 $L=2.0$ 和 $C=1.0$, 求谐振频率 f_0 (文件名 sy1_3.c)。

```
#include <stdio.h> // 包含标准输入输出头文件;
#include <math.h> // 包含数学头文件;
Main() // 经典 C 的主函数头部;
{ // 函数体开始;
    double L=2.0, C=1.0, f0; // 声明变量 L、C, f0 并赋初值;
    f0 = 1 / (2*3.1415926*sqrt(L*C)); // 用赋值语句计算谐振频率 f0;
    printf ( "f0 = %f\n", f0); // 调用标准输出函数 printf 输出 f0;
} // 函数体结束。
```

解: 编译预处理命令和语句的功能如程序右边所示, 用赋值语句计算和处理数据时用到数学函数, 如求平方根函数 sqrt, 在程序的开头的包含头文件中要包含 <math.h> 头文件。程序声明变量并赋初值, 输入电路参数 L、C 的数据, 用赋值语句计算谐振频率 f_0 , 用函数 printf 输出谐振频率 f_0 的值。源程序编译、运行后, 输出结果为:

```
f0 = 0.112540
```

4. 用宏定义设置符号常量

例 1.2.4 程序如下, 试分析宏代换的方法, 并写出程序运行后的结果 (文件名 sy1_4.c)。

```
#include <stdio.h> // 包含标准输入输出头文件;
#define N 6 // 定义符号常量 N;
#define M 2 // 定义符号常量 M;
main() // 经典 C 的主函数头部;
{ // 函数体开始;
    int iS; // 声明变量 iS;
    iS = N / M; // 用赋值语句计算 N / M;
    printf("%d\n", iS); // 调用标准输出函数 printf 输出 iS;
} // 函数体结束。
```

解: 宏定义命令定义符号常量 N 为 6, M 为 2, 在编译处理时对源程序中所有的符号常量 N、M 进行宏代换, 代换后 $iS = 6 / 2$ 运行程序时计算 iS 的值。源程序编译、运行后, 输出结果为: 3

5. 输入数据的提示

例 1.2.5 已知直角三角形的斜边长为 10, 一直角边长为 8, 求另一直角边之长 (文件名 sy1_5.c)。

```

#include <stdio.h>           // 包含标准输入输出头文件;
#include <math.h>            // 包含数学头文件, 程序中用到数学函数;
main()                      // 经典C的主函数头部;
{                            // 函数体开始;
    double a, b, c;          // 声明双精度变量 a、b、c;
    printf("Input c, a:\n"); // 在屏幕上显示待输入的变量 c, a;
    scanf("%lf, %lf", &c, &a); // 输入双精度变量的格式符用%lf;
    b=sqrt(c*c-a*a);         // 用数学函数平方根 sqrt() 计算另一直角边;
    printf ("b = %f\n", b);  // 调用标准输出函数 printf 输出数据;
}                            // 函数体结束。

```

解: 程序编译、运行时, 执行到 `scanf` 语句时, 编译器等候用户输入数据, 如果没有提示信息, 往往不知道应该输入什么数据, 因此, 在输入数据之前用 `printf` 函数提示用户输入数据是一个良好的习惯, 方便用户输入数据。源程序编译、运行后, 提示用户输入数据 Input c, a:

输入数据 10, 8 后回车, 屏幕显示输出结果:

```
b = 6.000000
```

1.2.2 编程方法

编程从主函数样板入手, 通过增加编译预处理命令和语句, 从简单到复杂, 逐步形成不同功能的程序。

1. 用经典主函数样板编写输出字符的程序

经典主函数样板由标准输入输出头文件、经典 C 的主函数头部和函数体组成。

```

#include <stdio.h>           // 标准输入输出头文件;
main()                      // 经典C的主函数头部;
{                            // 函数体开始;
    .....                  // 函数体;
}                            // 函数体结束。

```

函数体中至少包括一条输出语句, 在屏幕上显示输出信息, 一般用函数调用语句 `printf` 调用标准输出函数, 输出数据或字符串, 字符串后用 `\n` 表示换行。

例 1.2.6 编写输出字符串“用经典主函数样板编程”的源程序。

解: 编写输出字符串“用经典主函数样板编程”的源程序, 只要在主函数样板中加入 `printf` (“用经典主函数样板编程! \n”); 语句, 形成最简单的源程序 (文件名 `sy1_6.c`)。

```

#include <stdio.h>           // 包含标准输入输出头文件;
main()                      // 经典C的主函数头部;
{                            // 函数体开始;
    printf("用经典主函数样板编程! \n"); // 调用标准输出函数;
}                            // 函数体结束。

```

编译、运行程序输出结果如下:

用经典主函数样板编程!

2. 用 C89 或 C99 主函数样板编程

C89 或 C99 标准的主函数样板由标准输入输出头文件、主函数头部、函数体和 return 0; 语句组成。

```
#include <stdio.h>                                     // 标准输入输出头文件;
int main()                                              // C89 或 C99 的主函数头部;
{                                                       // 函数体开始;
.....                                                // 函数体;
return 0 ;                                             // 向操作系统返回 0 值, 表示程序正常退出;
}                                                       // 函数体结束。
```

例 1.2.7 编写输出字符串“用 C89 或 C99 主函数样板编程”和“注释是程序中的非执行部分”的源程序。

解：主函数返回值的类型是整型，返回值为 0 表示正常退出。在源程序中，可以加块注释和行注释说明程序功能和语句功能（文件名 sy1_7.c）。

```
/* 用块注释说明程序功能 */
#include <stdio.h>                                     // 包含标准输入输出头文件;
int main()                                              // C89 的主函数头部;
{                                                       // 函数体开始;
    printf("用 C89 或 C99 主函数样板编程!\n");        // 调用标准输出函数输出信息;
    printf("注释是程序的非执行部分\n");              // 调用标准输出函数输出信息;
    return 0;                                           // 向操作系统返回 0 值;
}                                                       // 函数体结束。
```

编译、运行程序输出结果如下：
用 C89 或 C99 主函数样板编程!
注释是程序的非执行部分

3. DEV-C++5.0 的函数样板

使用 DEV-C++5.0 集成开发环境编译运行程序时，输出窗口结束太快，看不清输出内容，习惯上在程序结束之前加一函数调用语句 getch();，也可调用 system("pause");暂停程序运行，DEV-C++5.4.0 以上的版本不需要加此语句，函数样板见表 1-11。

表 1-11 函数样板

DEV-C++5.0 加 getch();语句的函数样板	加 system("pause");语句的函数样板
<pre>#include <stdio.h> int main(void) { getch(); return 0; }</pre>	<pre>#include <stdio.h> #include <stdlib.h> int main(void) { system("pause"); return 0; }</pre>

例 1.2.8 用 DEV-C++5.0 集成开发环境编译运行程序，输出“Compiler DEV-C++\n”字符串。

解：用加 getch();语句的函数样板编制程序如下（文件名 sy1_8_1.c）。

```
#include <stdio.h> /* 包含标准输入/输出头文件 */
int main(void) /* 主函数由整型的函数类型、函数名和空参数组成 */
{ /* 函数体开始 */
    printf("Compiler DEV-C++\n"); /* 用格式输出函数 printf 输出字符串 */
    getch(); /* 暂停程序运行，查看输出内容 */
    return 0; /* 向操作系统返回 0 值，即程序正常退出 */
}
```

用加 `system("pause");` 语句的函数样板编制程序如下（文件名 `sy1_8_2.c`）。

```
#include <stdio.h> /* 包含标准输入/输出头文件 */
#include <stdlib.h> /* 包含 C 语言标准库头文件 */
int main(void) /* 主函数由整型的函数类型、函数名和空参数组成 */
{ /* 函数体开始 */
    printf("Compiler DEV-C++\n"); /* 用格式输出函数 printf 输出字符串 */
    system("pause"); /* 暂停程序运行，查看输出内容 */
    return 0; /* 向操作系统返回 0 值，即程序正常退出 */
} /* 函数体结束 */
```

4. 数据的输入和处理

输入数据时，可以采用声明变量时为变量赋初值的方法输入数据，可以调用 `scanf` 标准函数输入数据，还可以用宏定义方式将数据定义成符号常量输入数据。处理数据时，可以采用赋值语句简单的处理数据，然后用标准输出函数输出数据。

例 1.2.9 已知物体的质量 $m=2$ ，重力加速度 $g=9.8$ ，编程求物体所受的重力 f 。

解：根据重力公式， $f=m*g$ ，用三种方法输入数据，如表 1-12 所示。

表 1-12

声明变量时为变量赋初值	用标准函数输入数据	用宏定义输入数据
<pre>// 文件名: sy1_9_1.c #include <stdio.h> main() { float m=2,g=9.8f, f; f=m*g; printf("f=%f\n", f); }</pre>	<pre>//文件名: sy1_9_2.c #include <stdio.h> main() { float m,g,f; scanf("%f,%f",&m,&g); f = m * g; printf("f=%f\n",f); }</pre>	<pre>//文件名: sy1_9_3.c #include <stdio.h> #define m 2 #define g 9.8 main() { float f; f=m*g; printf("f=%f\n",f); }</pre>
$f = 19.600000$	$f = 19.600000$	$f = 19.600000$

5. 根据程序的逻辑顺序编程

例 1.2.10 将华氏温度 f 变换成摄氏温度 c ，计算公式为： $c=5*(f-32)/9$ ，若华氏温度 $f=50$ 度，试编程求摄氏温度 c 。

解：程序的逻辑顺序由变量声明、数据输入、数据处理和数据输出四条语句构成，先声明变量 c 和 f 为双精度型变量，调用标准输入函数 `scanf` 输入变量 f 的值 50.0，再用赋值语句计算摄氏温度 c 的值，然后调用标准输出函数 `printf` 输出 c 的数据（文件名 `sy1_10.c`）。