

计算机热门软件入门与提高丛书

掌握C语言，拥有更大舞台！

◎业内资深IT工程师带您轻松体验C语言学习之旅

◎科学的C语言学习路线图，风趣、幽默的讲解，

详实、扎实的内容结构，结合典型实例的深入

剖析都将使学习过程变得轻松、有趣！



1张DVD

800多分钟全程语音视频教学

+PPT教学课件+源代码+习题答案

C语言

入门与提高

陈锐 陈亚民 扶晓 编著



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

计算机热门软件入门与提高丛书

C语言

入门与提高

陈锐
陈亚民
扶晓
编著



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

C 语言是计算机专业的基础课和核心课程。本手册内容全面, 结构清晰, 语言通俗, 重难点突出, 所有程序都能够直接运行。本手册内容包括 C 语言概述、常用的程序开发环境、基本数据类型、表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、预处理命令、指针、结构图与联合体、位运算、文件、链表。

本手册可作为大中专院校的计算机及相关专业的 C 语言教材, 也可作为计算机软件开发、等级考试和软考人员的参考书。

本光盘为 15 个多小时的视频教学、PPT 电子课件、实例源代码与习题答案。

需要本手册或技术支持的读者, 请与北京清河 6 号信箱 (邮编: 100085) 发行部联系, 电话: 010-62978181 (总机) 转发行部、010-82702675 (邮购), 传真: 010-82702698, E-mail: tbd@bhp.com.cn。

C 语言入门与提高 / 陈锐, 陈亚民, 扶晓编著. 北京希望
电子出版社, 2011

(计算机热门软件入门与提高丛书)

ISBN 978-7-89499-140-9

I. C… II. ①陈… ②陈… ③扶… III. C 语言—程序
设计

责任编辑: 刘 蕊 / 责任校对: 马 君
责任印刷: 密 东 / 封面设计: 青青果园

北京希望电子出版社 出版

北京市海淀区上地三街 9 号金隅嘉华大厦 C 座 611

邮政编码: 100085

<http://www.bhp.com.cn>

北京市密东印刷有限公司印刷

北京希望电子出版社发行 各地新华书店经销

*

2011 年 2 月第 1 版

2011 年 2 月第 1 次印刷

印数: 1-3 000 册

开本: 787mm×1092mm 1/16

印张: 27.75

字数: 641 千字

定价: 49.80 元 (1 张 DVD)

C 语言是目前使用最为广泛的程序设计语言，也是计算机专业必修的一门课程，它是今后继续学习数据结构和算法的基础课程。C 语言功能丰富，表达能力强，使用灵活方便，程序执行效率高，移植性好，既具有高级语言的特点，又具有低级语言的特征。现在流行的 Visual C++ 和 C++ Builder 就是以 C 语言为基础的开发工具。

本手册全面地介绍了 C 语言的基本概念、基本语法及程序设计方法。本手册沿着学习程序设计的基本路线进行讲解，通过实例分析程序的结构和含义，每个实例都给出具体的代码，并配合流程图进行讲解。主要内容包括 C 语言概述、常用的程序开发环境、基本数据类型、表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、预处理命令、指针、结构图与联合体、位运算、文件、链表。学完本手册之后，读者基本可以熟练掌握程序设计的方法，并能熟练开发 C 语言程序。

本手册的特点

1. 内容全面，讲解详细

为了方便读者学习，本手册首先对 C 语言的特点及优势进行了讲解，然后讲解了各种 C 语言开发工具。本手册内容全面，覆盖了全部 C 语言的知识，对于每个知识点，都使用具体实例和生活中的实例进行讲解、对比，以便读者迅速掌握 C 语言。

2. 层次清晰，结构合理

本手册将 C 语言分篇、章、节和小节划分知识点，将知识点细化，层次清晰，结构合理，易于读者理解与学习。每一章、每一节、每一小节都被划分为单独的知识点，每一小节还可分为更多的知识点。在知识点的讲解过程中，循序渐进，由浅入深，先用为什么的方式引出概念，然后再进行讲解，最后通过例子强化知识点，这样的讲解方式使读者更容易理解和消化。

3. 结合图表，叙述简单

针对每个知识点，都结合图表和具体的实例去说明，将抽象的概念具体化，以便读者领会。在语言的叙述上，普遍采用短句子、易于理解的语言，而避免使用复杂句子和晦涩难懂的语言。通过以上方式的描述，读者可以更加容易学习并掌握 C 语言。

4. 例子典型，深入剖析

在讲解每一个 C 语言知识点时，结合具体例子进行剖析。在例子的选取上，都是一些最为常见且知识点丰富的典型程序。在每一章的最后或比较大的知识点后面，都给出了一个完整的程序，并对程序进行了具体的图解，深入地分析，在程序的最后，给出了运行结果。

5. 语言诙谐, 结合生活

在语言的讲解方面, 本手册采用诙谐的语言, 并将每个概念与现实生活结合对比, 从而使晦涩难懂的知识变得有趣, 从而让学习 C 语言程序设计易如反掌。通过本手册的学习, 可以非常容易地学会 C 语言, 使外行变内行。

6. 配有习题, 巩固知识

在每一章的最后, 都有一个小结, 对本章的知识点进行总结, 方便读者清楚本章的重点、难点。为了让读者熟练编写 C 语言, 本手册在每一章的最后都配有一定数量的实践题目, 在学习了每一章的内容之后, 可以通过这些习题试着编写 C 语言, 以巩固本章的学习内容。在本手册的光盘中, 提供了每一个例子的程序代码和课后习题代码。

本手册的内容

第 1 章: 如果读者还不知道该学习 C、C++、C# 还是 Java, 通过阅读本章, 将让你坚定学习 C 语言的决心。本章将告诉你 C 语言的特点, 为什么要学习 C 语言及基本的理论基础。

第 2 章: 本章对几种常用的 C 语言开发工具进行了详细的讲解。主要包括 Turbo C, Win-TC, LCC, Visual C++ 6.0, 可以选择一个自己喜欢的。

第 3 章: 本章主要介绍了变量和常量。首先讲解了为什么要有变量、为什么要有数据类型, 然后讲解了各种类型的变量定义及使用, 最后讲解了常量。

第 4 章: 本章主要讲解了运算符与表达式。首先介绍了运算符的分类, 什么是表达式, 然后讲解了常用的算术表达式、赋值表达式、关系表达式、逻辑表达式、逗号表达式。通过本章的学习, 读者将学会各种表达式的使用。

第 5 章: 本章主要讲解了顺序结构程序设计。首先介绍了语句的概念, 然后介绍了顺序结构程序的特点, 最后通过具体实例讲解如何编写顺序结构的程序。通过本章的学习, 读者将学会编写最简单的 C 语言程序——顺序结构的 C 语言程序。

第 6 章: 本章主要讲解了选择结构程序。首先讲解了条件运算符与条件表达式, 然后讲解了单分支的 if 选择语句、双分支的 if 选择语句、多分支的 if 选择语句、switch 选择语句, 最后通过具体实例结束本章的内容。通过本章的学习, 读者将进一步掌握较为复杂的程序设计——选择结构的 C 语言程序。

第 7 章: 本章主要讲解了循环结构程序设计。首先介绍了 3 种常用的循环语句——顺序结构的 C 语言程序 while 循环语句、do-while 循环语句、for 循环语句, 接着介绍了 goto 语句、break 语句和 continue 语句、多重循环结构的程序设计, 最后通过几个典型的实例结束本章的内容。通过本章的学习, 读者将掌握最复杂的程序结构——循环结构的 C 语言程序。

第 8 章: 本章主要讲解了数组。首先通过具体问题引出数组的概念, 然后讲解了一维数组、二维数组、字符数组。

第 9 章: 本章主要讲解了函数。首先讲解了函数的分类, 然后讲解函数的定义形式、局部变量和全局变量、函数的参数和函数的值、函数的调用、函数的嵌套调用、函数的递归调用、数组作为函数的参数, 最后通过几个实例结束本章的学习。通过学习本章的内容, 读者将掌握

模块化程序设计。

第 10 章：本章主要讲解了预处理命令。依次介绍了宏定义、文件包含命令、条件编译，通过学习本章的内容，读者可以编写出高效的程序并能提高程序的编译效率。

第 11 章：本章主要讲解了 C 语言的灵魂——指针。首先讲解了指针与地址的区别，然后讲解了指针变量、数组与指针、字符串与指针、指针数组与指向指针的指针、函数与指针、返回指针值的函数。指针是 C 语言最难的一个知识点，本章通过对比学习指针中容易混淆的概念，让读者加以区分。通过本章的学习，读者可以编写出高效的 C 语言程序。

第 12 章：本章主要讲解了结构体和联合体。首先讲解了结构体的定义、引用、初始化，然后讲解了结构体数组、指针与结构体、联合体、枚举类型。通过本章的学习，读者将能定义及使用比较复杂的数据，并能设计出比较实用的程序。

第 13 章：本章主要讲解了位运算。首先讲解了位运算符及位运算的优先级与结合性，然后介绍了各种位运算符，并利用一个典型实例讲解位运算符的使用方法。通过学习本章，读者将了解到计算机的底层操作。

第 14 章：本章主要讲解了文件。首先讲解了文件的分类，然后讲解了打开和关闭文件、读取文件和写入文件、文件的定位等操作，最后讲解了常见的其他文件操作函数。通过学习本章的内容，读者将掌握文件的操作。

第 15 章：本章主要讲解了链表。首先讲解了链表的概念、动态存储分配，然后将讲解了创建链表、插入链表、删除链表等操作，最后通过一个具体的实例结束链表的学习。通过学习本章，读者将掌握动态内存分配、最简单的数据结构。

本手册的第 1、3、4、6、7、8、9、11、15 章由陈锐编写，第 2、10、12 章由陈亚民编写，第 5、13、14 章由空军航空大学的扶晓编写。同时参与编写和资料整理的有王桂英、王金英、王红、陈小云、冉卫华、陈钰、王小艳、郑立红、米永刚、王天东、冯志祥、张小龙、王三中、卢香清、冀小幸、王佩、冯发成、董改香、王晓红、王胜军、刘红亮，在此一并表示感谢。

致谢

在此感谢北京希望电子出版社帮助本手册问世的所有人，尤其是编辑室主任杨如林先生。他十分看重本手册的应用价值，在他的努力下本手册才得以顺利出版，对此，深表感激。没有他们的帮助，这本手册就不可能与读者见面。

还要感谢我的导师张蕾教授。她是对我的职业生涯最有影响的人之一，她丰富的知识储备及敏锐的洞察力极大地影响了我的学习态度，促使我的学习能力和认识能力有了很大提高，为本手册的编著奠定了良好的知识基础。另外，耿国华教授在数据结构与算法领域给了我很大启发。

还要感谢我们的家人，在他们的默默付出与鼓励下，我才能顺利写完本手册。

最后感谢温县教育局电教馆的大力支持与帮助。

陈 锐

目录

第 1 篇 起步篇

第 1 章 C 语言概述	1
1.1 为什么要学习 C 语言	1
1.1.1 学习 C 语言的好处	1
1.1.2 如何学好 C 语言	3
1.2 C 语言的特点	3
1.3 程序设计基础	4
1.3.1 十进制数的表示	4
1.3.2 二进制数的表示	4
1.3.3 二进制数、十六进制数和 八进制数转换为十进制数	5
1.3.4 十进制数转换为二进制数	6
1.3.5 十进制数转换为十六进制数	8
1.3.6 十进制数转换为八进制数	9
1.3.7 二进制数与十六进制数、 八进制数的转换	10
1.3.8 计算机中的加减运算	11
1.4 小结	13
第 2 章 C 语言的开发环境	14
2.1 Turbo C 2.0 开发环境介绍	14
2.1.1 运行 Turbo C 2.0	14
2.1.2 Turbo C 2.0 环境设置	15
2.1.3 Turbo C 2.0 的使用	16
2.2 Win-TC 开发环境介绍	18
2.2.1 Win-TC 开发环境介绍	19
2.2.2 Win-TC 的使用	20
2.3 LCC 开发环境介绍	21
2.3.1 使用 LCC 开发环境创建项目	21
2.3.2 LCC 开发环境的介绍	23

2.3.3 LCC 的使用	24
2.4 Visual C++ 6.0 开发环境介绍	24
2.4.1 使用 Visual C++ 6.0 新建项目	25
2.4.2 使用 Visual C++ 6.0 新建源程序文件	26
2.4.3 使用 Visual C++ 6.0 开发环境	27
2.5 小结	28

第 2 篇 基础篇

第 3 章 变量和常量	29
3.1 变量	29
3.1.1 为什么要使用变量	29
3.1.2 变量究竟是什么—— 变量在计算机中的表示	30
3.1.3 如何定义变量	30
3.1.4 为什么要有数据类型	30
3.1.5 C 语言有哪些数据类型	31
3.1.6 什么符号可以作为变量名 ——用户定义的标识符	33
3.1.7 关键字——已经被占用的 标识符	33
3.1.8 初识变量赋值	34
3.2 整型变量	34
3.2.1 为什么要使用整型变量	35
3.2.2 基本整型变量	35
3.2.3 基本整型变量的输入 ——使用 scanf 函数	36
3.2.4 基本整型变量的输出	

——使用 printf 函数	37	4.2.3 自减表达式——--a 和 a--	66
3.2.5 长整型变量——解决		4.2.4 算术运算符的混合运算——	
大整数问题	38	算术运算符的优先级与结合性	68
3.2.6 短整型变量——节省内存空间	40	4.2.5 算术表达式的应用	69
3.2.7 无符号整型变量——		4.3 赋值运算符与赋值表达式	71
只包括非负的整数	41	4.3.1 赋值运算符与赋值表达式	71
3.2.8 整型变量的应用举例	42	4.3.2 类型转换	74
3.3 实型变量	44	4.3.3 赋值表达式的应用	76
3.3.1 为什么使用实型变量	45	4.4 关系运算符与关系表达式	77
3.3.2 单精度浮点型变量	45	4.4.1 关系运算符与关系表达式	77
3.3.3 双精度浮点型变量	47	4.4.2 关系表达式的应用	79
3.3.4 长双精度浮点型变量	49	4.5 逻辑运算符与逻辑表达式	79
3.3.5 实型变量的应用	49	4.5.1 逻辑运算符与逻辑表达式	80
3.4 字符型数据类型	51	4.5.2 逻辑表达式的应用	81
3.4.1 为什么要使用字符型变量	51	4.6 逗号表达式	83
3.4.2 字符型变量	51	4.6.1 逗号运算符与逗号表达式	83
3.4.3 字符型数据的输入与输出		4.6.2 逗号表达式中应注意的问题	83
——使用 %c 和 %s	52	4.6.3 逗号表达式的应用	84
3.4.4 字符型变量的应用	53	4.7 小结	84
3.5 常量	55	习题	84
3.5.1 为什么使用常量	55	第 5 章 顺序结构程序设计	86
3.5.2 整型常量	56	5.1 语句和程序	86
3.5.3 实型常量	56	5.1.1 什么是简单语句	86
3.5.4 字符型常量与字符串型常量	56	5.1.2 为什么使用复合语句	87
3.5.5 常量与常数	57	5.1.3 什么是复合语句	88
3.6 小结	57	5.1.4 使用复合语句应注意的问题	89
习题	57	5.1.5 为什么要有程序	90
第 4 章 运算符与表达式	59	5.1.6 C 语言程序的框架结构	90
4.1 表达式	59	5.2 顺序结构程序设计	91
4.1.1 运算符有哪些	59	5.2.1 顺序结构程序设计的特点——	
4.1.2 什么是表达式	60	自上而下执行每条语句	91
4.1.3 运算符的优先级与结合性	60	5.2.2 顺序结构程序设计 (一)	
4.1.4 数据类型转换	61	——求圆的面积	92
4.2 算术运算符与算术表达式	62	5.2.3 顺序结构程序 (二) ——	
4.2.1 基本算术表达式	62	求一元二次方程	
4.2.2 自增表达式——++a 和 a++	64	$ax^2+bx+c=0$ 的根	94

5.2.4 顺序结构程序设计(三)—— 将小写字母转换为大写字母	95	7.2.1 while 循环语句——当型循环	136
5.3 小结	97	7.2.2 while 循环语句的应用	138
习题	97	7.3 do-while 循环语句	139
第 6 章 选择结构程序设计	98	7.3.1 为什么要有 do-while	139
6.1 条件运算符	98	7.3.2 do-while 循环语句—— 直到型循环	139
6.1.1 为什么要有条件运算符	98	7.3.3 比较 while 循环结构和 do-while 循环结构	141
6.1.2 条件运算符与条件表达式	99	7.4 for 循环语句	144
6.2 if 选择语句	102	7.4.1 for 循环语句	144
6.2.1 为什么要有 if 选择语句 ——理解容易	102	7.4.2 for 循环语句的灵活性	147
6.2.2 if 选择语句——单分支的 选择结构	102	7.4.3 几种循环语句的比较	152
6.2.3 if-else 选择语句——双分支 的选择结构	106	7.5 goto 循环语句	153
6.2.4 if-else if-...-else 选择语句 ——多分支选择结构	108	7.5.1 goto 语句——无条件 转移语句	153
6.2.5 if 选择语句的嵌套	112	7.5.2 goto 语句构成的循环语句 ——向前跳转	153
6.2.6 设置标志变量	116	7.5.3 goto 语句是一种不常用的 结构——与 while 语句、for 语句的比较	154
6.2.7 if 选择语句与选择运算符 ——相互转换	117	7.6 break 语句	156
6.3 switch 选择语句	118	7.6.1 循环中的“提前开溜”	156
6.3.1 switch 选择语句	118	7.6.2 break 语句	156
6.3.2 switch 选择语句的应用	119	7.7 continue 语句	157
6.4 选择结构程序设计的应用	125	7.7.1 为什么要有 continue 语句	157
6.5 小结	133	7.7.2 continue 语句	158
习题	133	7.7.3 continue 语句与 break 语句的区别	159
第 7 章 循环结构程序设计	134	7.7.4 continue 语句的应用	160
7.1 为什么要有循环结构	134	7.8 多重循环程序设计	161
7.1.1 如何重复输出多个 hello world——使用 循环可以少输入代码	134	7.8.1 为什么要有循环的嵌套 ——一层循环是不能解决 所有问题的	161
7.1.2 如何求连续的 n 个自然数的 和——循环代码可以重复使用	135	7.8.2 循环的嵌套——循环结构 中还可以有循环结构	162
7.2 while 循环语句	136	7.8.3 多重循环的应用——输出 乘法口诀表	163

7.9 循环结构程序应用举例	166
7.9.1 循环结构程序（一）—— 求 π 的近似值	166
7.9.2 循环结构程序（二）—— 求 100~200 之间的所有质数	169
7.9.3 循环结构程序（三）—— 求 100~200 之间的所有 质数的改进	171
7.9.4 循环结构程序（四）——百鸡 问题	172
7.10 小结	174
习题	174

第 3 篇 进阶篇

第 8 章 数组

8.1 为什么要有数组	175
8.1.1 多变量的解决之路—— 数组的引入	175
8.1.2 初识数组	176
8.1.3 数组的数组——维度的出现	177
8.2 一维数组	179
8.2.1 一维数组的定义	179
8.2.2 一维数组的引用—— 其实很简单	181
8.2.3 一维数组的初始化—— 方法有许多	182
8.2.4 一维数组的应用举例—— 求数组各元素的平均值	183
8.2.5 一维数组的应用—— 数组元素的倒排	185
8.2.6 一维数组的应用—— 冒泡排序	187
8.3 二维数组	190
8.3.1 二维数组的定义	190
8.3.2 二维数组的引用—— 其实并不难	191
8.3.3 二维数组的初始化——	

多样的初始化	192
8.3.4 二维数组的应用—— 计算各科成绩的平均分	195
8.3.5 二维数组的应用—— 矩阵的转置	196
8.3.6 二维数组的应用—— 有趣的魔方阵	198
8.4 字符数组	201
8.4.1 字符数组的定义	201
8.4.2 字符数组的初始化	202
8.4.3 字符数组的连续输出 ——格式符 %s	205
8.4.4 字符数组的连续输入 ——格式符 %s	205
8.4.5 常用的字符串处理函数	206
8.4.6 字符数组的应用	210
8.5 小结	213
习题	214

第 9 章 函数

9.1 函数及分类	215
9.1.1 为什么要有函数	215
9.1.2 函数有哪些——库函数 与用户函数	216
9.2 函数的定义形式	217
9.2.1 不带参数函数的定义	217
9.2.2 带参数函数的定义—— 有参数列表	218
9.2.3 带参数函数的定义—— 形参定义的传统方式	218
9.3 局部变量与全局变量	219
9.3.1 局部变量	219
9.3.2 全局变量	221
9.4 函数的参数与函数的值	223
9.4.1 形式参数与实际参数	223
9.4.2 函数的返回值	225
9.5 函数的调用	228
9.5.1 函数调用的一般形式	228

9.5.2 函数调用的方式	230
9.5.3 对被调用函数的声明	230
9.6 函数的嵌套调用	233
9.6.1 函数的嵌套调用过程	233
9.6.2 函数的嵌套应用——求 1~n 自然数的 k 次方之和	234
9.7 函数的递归调用	235
9.7.1 什么是递归——自己 调用自己	235
9.7.2 递归函数的应用——求 n!	236
9.7.3 递归函数调用的应用 ——Fibonacci 数列	238
9.8 数组作为函数的参数	239
9.8.1 数组元素作为函数的 参数——只能传递一个 数组元素	239
9.8.2 数组名作为函数的参数—— 可以传递整个数组	241
9.8.3 数组名作为参数传递—— 传递的是数组的首地址	244
9.8.4 多维数组名作为函数的 参数——传递整个数组	247
9.9 变量的存储	250
9.9.1 什么是动态存储与静态 存储——生存期不同	250
9.9.2 auto 变量——自动 类型变量	250
9.9.3 static 变量——静态变量	250
9.9.4 register 变量——使运行 速度更快的变量	253
9.9.5 extern 变量——声明 外部变量	254
9.10 内部函数与外部函数	256
9.10.1 内部函数——使用 static	256
9.10.2 外部函数——使用 extern	257
9.11 函数应用举例	258
9.11.1 函数应用——递归求解 n 个数的最大值	258

9.11.2 函数应用——递归求解 a 与 b 的最大公约数	260
9.11.3 函数应用——字符数组 元素逆序存放	261
9.12 小结	262
习题	262

第 10 章 预处理命令

10.1 宏定义	263
10.1.1 为什么要有宏定义	263
10.1.2 不带参数的宏定义	264
10.1.3 宏定义与常量定义的区别	265
10.1.4 带参数的宏定义	265
10.1.5 宏定义中参数与函数 参数的区别	266
10.2 文件包含命令	268
10.2.1 为什么要有文件包含命令	268
10.2.2 文件包含命令——#include	268
10.3 条件编译	270
10.3.1 第一种条件编译命令 ——#ifdef	270
10.3.2 第二种条件编译命令—— #ifdef...#elif...#endif	272
10.3.3 第三种条件编译命令 ——#ifndef	273
10.3.4 第四种条件编译命令——#if	274
10.4 小结	274
习题	275

第 11 章 指针

11.1 指针的相关概念	276
11.1.1 为什么要有指针和指针变量	276
11.1.2 什么是地址	276
11.1.3 什么是指针变量—— 存放地址的变量	277
11.2 指针变量	278
11.2.1 定义指针变量	278
11.2.2 使用指针变量	280

11.2.3 指针变量作为函数参数	283	12.2.2 定义结构体数组	334
11.3 数组与指针	287	12.2.3 结构体数组的初始化	335
11.3.1 指向数组元素的指针	287	12.2.4 结构体数组的应用	336
11.3.2 通过指针引用数组元素	288	12.3 指针与结构体	339
11.3.3 指针变量的自增运算和 自减运算	292	12.3.1 指向结构体变量的指针	339
11.3.4 数组指针作为函数的参数	292	12.3.2 指向结构体数组的指针	341
11.3.5 指向多维数组的指针变量	298	12.3.3 结构体变量和指向结构体 的指针作为函数的参数	342
11.4 字符串与指针	304	12.4 typedef 类型定义	346
11.4.1 指向字符串的指针变量	304	12.4.1 typedef——为数据类型 重新起个名字	346
11.4.2 字符串指针作为函数的参数	310	12.4.2 使用 typedef 应 注意的问题	347
11.4.3 字符数组与字符指针 变量的比较	313	12.4.3 typedef 的应用	348
11.5 指针数组与指向指针的指针	314	12.5 联合体	351
11.5.1 指针数组	314	12.5.1 为什么要有联合体—— 节省内存, 多选一	351
11.5.2 指向指针的指针	317	12.5.2 定义联合体	352
11.5.3 指针数组作为 main 函数的参数	319	12.5.3 引用联合体变量	353
11.6 函数与指针	320	12.5.4 使用联合体应该注意的问题	353
11.6.1 函数指针调用函数	320	12.5.5 联合体的应用	354
11.6.2 指向函数的指针作为 函数参数	321	12.6 枚举类型	357
11.7 返回指针值的函数	323	12.6.1 为什么要有枚举类型—— 变量的值只有有限的几种	357
11.7.1 为什么函数要返回指针值 ——可以返回多个值	323	12.6.2 定义枚举类型及变量	358
11.7.2 返回指针类型的函数	324	12.6.3 使用枚举类型应注意的问题	358
11.8 小结	325	12.6.4 枚举类型的应用	359
习题	326	12.7 小结	360
第 12 章 结构体与联合体	327	习题	360
12.1 结构体	327	第 13 章 位运算	361
12.1.1 为什么要有结构体	327	13.1 为什么要有位运算与位运算符	361
12.1.2 结构体类型的定义	328	13.1.1 为什么要有位运算 ——高效率的运行	361
12.1.3 定义结构体变量	328	13.1.2 位运算符	361
12.1.4 引用结构体变量	330	13.1.3 位运算符的优先级与结合性	362
12.1.5 结构体变量的初始化	332	13.2 位运算符与位运算	362
12.2 结构体数组	334	13.2.1 按位与运算符和按位与运算	362
12.2.1 为什么要有结构体数组	334		

第 14 章 文件 374

第4篇 提高篇

第 15 章 链表..... 395

15.2.5 链表的综合操作	413
15.3 链表操作的应用	414
15.3.1 链表操作的应用—— 逆置链表	415
15.3.2 链表操作的应用—— 约瑟夫问题	420
15.4 小结	423
习题	423

附 录	424
附录 1 程序流程图	424
附录 2 常用字符与对应的 ASCII 码对照表	427
附录 3 运算符的优先级与结合性	428
附录 4 常用函数	429

第 1 篇 起步篇

第 1 章 C 语言概述



本章导读

C语言是一门经典且优秀的语言。C语言是1972年由美国贝尔实验室的Dennis Ritchie在B语言的基础上设计发明的，并首次用C语言改写了UNIX操作系统。随后C语言被使用在多种计算机上，C语言同时得到广泛使用，从此风靡全球。因此，C语言出现了多种版本。后来，美国国家标准研究所ANSI为C语言制定了一个统一的标准，成为现行的C语言标准。本章主要介绍的内容包括为什么要学习C语言、C语言的特性及程序设计基础。



学习要点

学习C语言的好处	如何学好C语言
C语言的特点	十进制数的表示
二进制数的表示	二进制数、十六进制数和八进制数转换为十进制数
十进制数转换为二进制数	十进制数转换为十六进制数
十进制数转换为八进制数	二进制数与十六进制数、八进制数的转换
计算机中的加减运算	

1.1 为什么要学习 C 语言

C语言经历了30多年仍然经久不衰，最重要的是它具有强大的功能，与其他语言相比具有一定的先天优势。在学习C语言之前，首先要告诉大家的是，为什么要学习C语言，也就是学习C语言有哪些好处，以及如何学好C语言。

1.1.1 学习 C 语言的好处

1. C语言是各种语言的基础

目前最为流行的语言有C语言、C++语言、Java语言、C#等，而C语言是所有这些语言的基础。可以说，现在的程序员没有哪个不知道C语言的鼎鼎大名的，他们学习的第一门语言基本上也都是C语言。C语言与其他语言相比，具有以下特色。

(1) 如果有了C语言作为程序设计语言的基础，你就会很容易地掌握其他编程语言如Java、C++、C#，因为它们都是以C语言作为基础的语言。

(2) 相比较其他编程语言如Java、Basic、C#，C语言是一门低级语言，而学习低级语言能

够使你更加清楚地了解计算机的工作原理。

(3) C语言通过阐述指针的概念,使你更加清楚地了解到计算机操作的本质。而C#和Java则没有这个概念,因此难以领略到指针的强大和语言的本质。

(4) 因为C语言是一门比较基础的语言,更接近硬件,所以运行效率非常快。

(5) C语言允许直接访问物理地址,可以进行位一级的运算,能实现汇编语言的大部分功能,因此C语言又是一门介于高级语言和汇编语言之间的语言。

(6) 与汇编语言相比,C语言的可移植性好。

正是C语言的这些特点,你会在今后学习的过程中,掌握到C语言及计算机的一些内部的工作原理,会对计算机有更加深刻的认识。

2. C语言是一种主流的语言

你可能经常听人说到像Visual C++、Visual Basic、Eclipse、Visual C#等开发工具,这些都是比较流行的软件开发工具。你要想成为一名优秀的程序员,需要掌握以上的软件开发工具,要使用这些软件开发工具,C语言则是使用这些软件开发工具的必备基础,而目前比较流行的软件开发工具都是以C语言为基础的。因此,C语言则理所当然成为主流的程序设计语言。它的主流地位主要体现在以下几个方面。

(1) C语言从诞生到现在已经有将近40年的历史,因此给我们留下了相当多的C语言现成可用的代码,这些代码都是比较宝贵的资源,另外有相当多的人仍然在支持并使用C语言。

(2) 设备驱动程序和操作系统只能用C语言编写。如果你根本不熟悉C语言,该如何编写和修改它们呢?

(3) 现在比较流行的计算机等级考试、软件水平考试都以C语言作为主流语言。如果你准备参加这些考试和一些培训机构的辅导,C语言是一个非常好的选择。

(4) 程序设计语言是今后要学习的数据结构课程的基础,数据结构中绝大部分都是采用C语言描述。

3. 与C语言相关的工作比较多

为了以后找工作打算,你最好选择C语言。如果选择了C语言,在今后找工作过程中发现有以下好处。

(1) 好多软件公司都招聘精通C语言、C++语言的Visual C++程序员。如果你比较擅长Visual C++,会很容易找到相应的工作。

(2) 任何涉及到硬件开发的设备都支持C语言,如手机开发、嵌入式开发。

哇,原来学习C语言有这么多的好处!那还等什么,马上开始学习C语言吧。既然现在你已经坚定了学习C语言的信心,那就让我来告诉你如何学好C语言吧。

1.1.2 如何学好C语言

那如何才能学好C语言呢?你可能是一名刚接触编程语言爱好者,首先要克服心理上的障

碍，不要放弃，就一定能学好 C 语言。要学好 C 语言，你必须做到以下几点。

(1) 首先你必须明白字符、ASCII 码等基本概念，在学习的过程中，要记住一些特殊字符与 ASCII 码的转换关系。

(2) 要非常清楚地知道在计算机内部，所有的数据的存储和传输都是使用二进制数，因此要掌握好二进制数与十进制数、十六进制数、八进制数的转换。

(3) 在学习的过程中，要记住 C 语言中的 32 个关键字。不过不要专门去记这些东西，在学习的过程中，你会自动就记住了。

(4) 要记住 C 语言中运算符的优先级，这一点可是非常重要的。

(5) 要真正地学好 C 语言中的顺序、选择和循环 3 种基本程序结构，还要掌握 `break` 语句和 `continue` 语句。

(6) 要熟练掌握指针的用法，指针是 C 语言中的最为经典的部分，也是与其他语言不同的地方，掌握好了指针，才能在今后的程序设计中少犯错误。

(7) 掌握一些比较简单的算法，例如，冒泡排序、插入排序、魔方阵等。

(8) 程序设计是一门实践性比较强的课程，因此，要真正地学好 C 语言，必须要多上机实践，这是学好 C 语言最重要的一条。

(9) 在学习 C 语言的过程中，要学会使用离散的思维思考问题。计算机处理的问题都是二进制方面的问题，所处理的数据是不连续的，这与我们在中学学习数学可是不一样的，需要大家在学习的过程中慢慢体会。

只要你有足够的信心和恒心，多上机编程，相信一定能够把 C 语言学习好。

1.2 C 语言的特点

C 语言之所以具有如此强大的生命力，并且成为最受欢迎的语言，除了它优于其他编程语言的先天优势之外，C 语言本身也具有一些内在的特点深受人们喜爱。C 语言具有以下特点。

(1) C 语言简洁、书写形式自由，实现同一个功能的语句可以有多种的写法。例如，在 `for` 循环语句中，可以将赋初值放在 `for` 语句的前面，也可以在 `for` 语句内部。循环的判断条件还可以省略，循环变量的增值可以在循环条件中，也可以在循环语句的内部。

(2) C 语言是结构化的语言，具有顺序、选择和循环 3 种常用的控制结构。C 语言用函数作为程序的模块单位，从而使程序层次清晰，便于调试和维护。

(3) C 语言语法检查不太严格，给予了程序设计者更大的自由空间。

(4) C 语言是一门基础的编程语言。C 语言本身既是一门“高级语言”，但是同时也具有低级语言的特点，因此称它为基础语言。C 语言是一门“高级语言”，是因为它的书写形式更接近自然语言；C 语言是一门“低级语言”，是因为它能对“位”进行操作，可以实现汇编语言的大部分功能，可以对硬件进行操作。

(5) C 语言生成的目标代码质量高，程序的执行效率高。

(6) C 语言的可移植性好。