

面向 21 世纪

高等学校系列教材

新编 C 语言程序设计教程

(第二版)

New C Language Programming Course

周学毛 编著

西安电子科技大学出版社
<http://www.xduph.com>

面向 21 世纪高等学校系列教材

新编 C 语言程序设计教程

(第 二 版)

周学毛 编著

西安电子科技大学出版社

2004

内 容 简 介

C 语言是一种应用广泛的程序设计语言。

本教材易读好懂，是一本 C 语言及程序设计的特色教材，不要求读者具有其它程序设计语言基础。

本教材分入门、进阶及提高三个部分，全面介绍了 C 语言的基本语法，标准数据类型、扩展数据类型及数据描述方法，C 语言语句及操作描述方法，C 语言程序设计方法，常用的基本算法和编程技巧，并拓展介绍了 C++ 语言与面向对象的程序设计方法。

本教材以 C 语言程序设计为主线，强调三有：有问题、有方法、有实现，内容选取科学、全面、实用，文字表达简明、通俗、易懂，例题、习题选择丰富、多样，难点、重点分析深入、透彻。

本教材可作为计算机及相近专业“C 语言程序设计”课程的教材，亦可作为“C 语言程序设计”的培训、自学用教材。

★本书配有电子教案，需要者可与出版社联系，免费提供。

图书在版编目 (CIP) 数据

新编 C 语言程序设计教程 / 周学毛编著. —2 版. —西安: 西安电子科技大学出版社, 2004.11
(面向 21 世纪高等学校系列教材)

ISBN 7-5606-0865-5

I. 新… II. 周… III. C 语言—程序设计—高等学校教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 098904 号

策 划 马乐惠

责任编辑 阎 彬 马乐惠

出版发行 西安电子科技大学出版社 (西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http: //www.xduph.com

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 陕西华沐印刷科技有限责任公司

版 次 2000 年 7 月第 1 版 2004 年 11 月第 2 版 2004 年 11 月第 7 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 20.75

字 数 490 千字

印 数 30 001~34 000 册

定 价 22.00 元

ISBN 7-5606-0865-5 / TP·0452(课)

XDUP 1136012-7

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜，谨防盗版。

第二版前言

本教材的编写以作者 2000 年编著的《新编 C 语言程序设计教程》为基础,融入了这几年 C 语言及程序设计的教学研究与实践。希望这次的编写能给“C 语言程序设计”的教材建设锦上添花。

这次的编写充分保留了第一版的特色,继续强调学 C 语言是学方法、学技能,是学好、用好计算机的重要基础;继续强调以程序设计为中心、为主线,学程序设计语言只是手段,学程序设计、学计算机解决问题的方法才是目的;继续强调三有:有问题、有方法、有实现;继续强调简单明了,分散难点,易读好懂。

这次的编写对原有的体系架构基本未作调整,因为这已经得到了读者的高度认可,依然包括入门、进阶、提高三个部分,共计 14 章。本教材在内容上将函数提前,指针推后,以使读者快速进入 C 语言程序设计世界,相对降低教学难度;对内容进行了全面优化处理,进行了大量改写,讲述更加简明扼要,例题和习题更加丰富、更加典型;还增加了图形处理、Windows 编程基础等内容。

入门部分为第 1~7 章,是整个 C 语言程序设计的基础篇和重点篇。这部分详细介绍了 C 语言的语法成分、C 语言程序的结构、基本数据类型与数据描述方法、函数、C 语言语句与操作描述方法。通过这部分的介绍,可使读者基本掌握 C 语言的语法,了解结构化程序设计方法,能动手编制简单的 C 语言程序。

进阶部分为第 8~12 章,为整个 C 语言程序设计的关键篇和中心篇。这部分主要对数据类型进行扩充,对数据描述方法进行扩充,全面介绍数组、结构体、指针及文件等用户自定义类型的使用,同时介绍一些常用的算法及编程技巧。这部分通过大量的典型实例让读者基本掌握结构化程序设计方法,能编制一般的 C 语言程序。

提高部分为第 13~14 章,是 C 语言程序设计的延伸篇。这部分内容包含两个方面:一方面在 C 的基础上介绍 C++,介绍面向对象的程序设计方法,同时从 C++的角度看 C;另一方面介绍三个 C 语言程序设计的综合实例,并以此圆满结束 C 语言的学习。通过这部分的介绍,可使读者了解面向对象的程序设计方法,进一步提高 C 语言程序设计能力。

对于初学程序设计的读者,当初作者有一点想法,这里再次建议读者一定要多读程序,一定要多动手编写程序,边学边做,边做边学,小题大作。程序设计的學習强调先模仿,在模仿的基础上改进,在改进的基础上提高,同时建议最终能进入 VC++ 的程序设计世界。

本教材若作为第一语言使用,建议安排 48~64 学时;若作为第二语言使用,建议安排 32~48 学时。第三部分可根据需要以讲座的形式进行或安排自学。

非常感谢这些年使用第一版《新编 C 语言程序设计教程》的几万读者,也期待本教材能给读者带来更多的惊喜与收获,同时也真诚希望读者能提供使用本教材的宝贵意见与建议。

作者

2004 年 8 月于岳麓山

第一版前言

C 语言是一种应用广泛的程序设计语言。“C 语言程序设计”课程是计算机及相近专业的一门重要必修专业课程，学 C 语言是一门技能，学一门本领。学好 C 语言，是学好、用好计算机的重要基础。

在这个已经诞生了多种版本 C 语言程序设计教材的年代，作者怀着几多希望与几多忧虑，在总结自身多年教学经验及教改实践的基础上，参考同仁编写的多本 C 语言教材，编著了这本《新编 C 语言程序设计教程》教材。

本教材不要求读者具有其它程序设计语言基础。在本教材的编写过程中，作者高度遵循如下原则：

- 以 C 语言程序设计为主线，学程序设计语言只是手段，掌握程序设计的方法才是目的。

- 内容选取力求科学、全面、实用，强调基础，强化应用。全面强调三有：有问题、有方法、有实现。

- 文字简明，表达力求通俗、易懂；例题、习题前呼后应，选择力求丰富、多样；难点、重点分散，分析力求深入、透彻。

本教材分入门、进阶、提高三个部分，共 14 章。体系安排上将函数提前，将指针推后，以使读者快速进入 C 语言程序设计，相对降低教与学的难度。

入门部分包括第 1~7 章，为整个 C 语言程序设计的基础篇、重点篇，详细介绍 C 语言的语法成分、C 程序的结构、基本数据类型与数据描述方法、程序的三种基本结构（顺序、选择、循环）、函数、C 语言语句与操作描述方法。通过这部分的介绍，让读者基本掌握 C 语言的语法，了解结构化程序设计方法，能动手编制简单的 C 程序。

进阶部分包括第 8~12 章，为整个 C 语言程序设计的关键篇、中心篇，主要对数据类型进行扩充，对数据描述方法进行扩充，全面介绍数组、结构体、指针及文件等用户自定义类型的使用，同时介绍一些常用的算法及编程技巧，通过大量的典型实例让读者基本掌握结构化程序设计方法，能编制一般的 C 语言程序。

第 13、14 章为提高部分，是 C 语言程序设计的延伸。一方面从 C 到 C++，在 C 的基础上介绍 C++，介绍面向对象的程序设计方法，同时从 C++ 的角度看 C；另一方面介绍两个 C 语言程序设计的综合实例，并以此圆满地结束 C 语言的学习。通过这部分的介绍，使读者了解面向对象的程序设计方法，进一步提高 C 语言程序设计能力。

为方便读者，本教材还提供了常用标准函数等附录供读者参考。

对于初学程序设计的读者，作者有一点想法和建议：建议读者一定要多读程序，一

定要多动手编程序。学习程序的一般规律是：先模仿，在模仿的基础上改进，在改进的基础上提高。同时要学会“小题大做”，每章提供的习题虽然都是一些小问题，但要按程序设计的一般要求去做，这样有利于养成良好的程序设计风格。程序设计教学强调边讲边做，边学边做，举一反三。

若作为第一语言使用本教材，建议安排 64 学时；若作为第二语言，建议安排 48 学时。实践部分建议安排 16~32 学时，为强化实践环节，建议设置半周的大型作业或一周的课程设计。第三部分内容可根据需要以讲座的形式进行或安排自学。

作者的同事李根强老师参加了第 8~11 章初稿的编写工作。

在本教材的编写过程中，作者参考了大量的资料，吸取了多位同仁的经验，借此诚致万分谢意。作者还要特别感谢本书责任编辑马乐惠女士的大力支持和辛勤劳动。

作者曾以善良的眼光寻找他人教材的缺陷与不足，诚恳希望读者提供使用本教材的宝贵意见与建议。

本教材配有学习指导书及电子教案。需要电子教案的读者请与西安电子科技大学出版社联系，免费索取。

作 者

1999 年 12 月于岳麓山

目 录

第一部分 C语言程序设计入门篇

第1章 C语言基础.....3	2.5 标准函数.....20
1.1 C语言概述.....3	2.6 量的定义方法.....20
1.1.1 C语言的产生与发展.....3	2.6.1 常量定义.....20
1.1.2 C语言的特点.....4	2.6.2 变量定义.....21
1.2 基本语法成分.....5	习题二.....23
1.2.1 字符集.....5	第3章 表达式.....25
1.2.2 标识符.....5	3.1 表达式基础.....25
1.2.3 关键字.....6	3.1.1 表达式的组成.....25
1.2.4 运算符.....6	3.1.2 表达式的书写.....26
1.3 C语言程序的结构.....6	3.1.3 表达式的分类.....26
1.3.1 C语言程序的结构.....8	3.1.4 表达式的计算.....27
1.3.2 C语言程序的书写.....9	3.2 算术表达式.....29
1.4 C语言程序的实现.....9	3.3 赋值表达式.....30
1.4.1 实现步骤.....9	3.3.1 赋值运算.....31
1.4.2 程序的执行.....10	3.3.2 赋值类型转换.....31
1.4.3 Turbo C上机操作.....10	3.3.3 复合赋值运算.....32
习题一.....12	3.4 逻辑表达式.....33
第2章 基本数据类型.....13	3.4.1 关系表达式.....33
2.1 整型.....13	3.4.2 逻辑表达式.....34
2.1.1 基本整型.....13	3.5 位运算表达式.....35
2.1.2 拓展整型.....14	3.5.1 位逻辑运算.....35
2.2 浮点型.....15	3.5.2 移位运算.....36
2.2.1 单精度型.....16	3.6 其它表达式.....37
2.2.2 双精度型.....16	3.6.1 条件表达式.....37
2.3 字符型.....17	3.6.2 逗号表达式.....38
2.3.1 字符型数据.....17	3.6.3 其它运算.....38
2.3.2 字符串数据.....18	习题三.....39
2.4 逻辑类型.....19	第4章 顺序结构程序设计.....42
2.4.1 逻辑类型.....19	4.1 C语言语句.....42
2.4.2 逻辑运算.....19	4.1.1 控制语句.....42

4.1.2 表达式语句.....	42	6.4.1 continue 语句.....	86
4.1.3 特殊语句.....	43	6.4.2 break 语句.....	87
4.2 数据输入.....	44	6.5 多重循环.....	88
4.2.1 一般格式.....	45	6.6 程序设计举例.....	90
4.2.2 格式字符.....	45	习题六.....	96
4.2.3 说明.....	46	第 7 章 函数	99
4.3 数据输出.....	47	7.1 函数的定义.....	99
4.3.1 一般格式.....	47	7.1.1 函数的结构.....	99
4.3.2 格式字符.....	47	7.1.2 返回语句.....	100
4.3.3 说明.....	48	7.1.3 函数的定义.....	101
4.3.4 交互信息.....	49	7.2 函数的调用.....	104
4.4 算法与程序设计.....	50	7.2.1 函数调用的一般形式.....	104
4.4.1 算法.....	50	7.2.2 函数调用的方式.....	105
4.4.2 程序.....	50	7.2.3 被调函数说明.....	107
4.4.3 NS 流程图.....	51	7.3 数据传递方法.....	108
4.4.4 结构化程序设计方法.....	51	7.4 嵌套调用和递归调用.....	109
4.4.5 程序设计过程.....	52	7.4.1 嵌套调用.....	109
4.5 程序设计举例.....	52	7.4.2 递归调用.....	111
习题四.....	57	7.5 变量的作用域.....	113
第 5 章 选择结构程序设计	59	7.5.1 局部变量.....	114
5.1 if 语句.....	59	7.5.2 全局变量.....	114
5.1.1 单分支 if 语句.....	59	7.5.3 标识符的作用域.....	115
5.1.2 双分支 if 语句.....	60	7.6 变量的存储类别.....	116
5.1.3 多分支 if 语句.....	63	7.6.1 变量的存储类别.....	116
5.2 switch 语句.....	64	7.6.2 局部变量的存储类别.....	116
5.3 goto 语句.....	66	7.6.3 全局变量的存储类别.....	118
5.4 程序设计举例.....	67	7.6.4 变量小结.....	119
习题五.....	74	7.7 结构化程序设计方法.....	122
第 6 章 循环结构程序设计	77	7.7.1 程序设计的基本手段.....	122
6.1 while 语句.....	77	7.7.2 模块化程序设计.....	122
6.2 do-while 语句.....	80	7.7.3 结构化程序的书写和阅读.....	123
6.3 for 语句.....	82	习题七.....	129
6.4 终止循环语句.....	86		

第二部分 C 语言程序设计进阶篇

第 8 章 数组类型	135	8.1.1 一维数组的定义.....	135
8.1 一维数组.....	135	8.1.2 一维数组的引用.....	136

8.1.3 一维数组的初始化.....	137	10.1.4 指针作函数参数.....	194
8.1.4 一维数组作函数参数.....	139	10.2 指针与数组.....	195
8.2 二维数组与多维数组.....	141	10.2.1 指向一维数组的指针变量.....	195
8.2.1 二维数组的定义.....	141	10.2.2 数组作函数参数.....	197
8.2.2 二维数组的引用.....	142	10.2.3 指向二维数组的指针变量.....	200
8.2.3 二维数组的初始化.....	142	10.3 指针与字符串.....	202
8.3 字符数组与字符串.....	145	10.3.1 字符串的指针表示.....	202
8.3.1 字符数组.....	145	10.3.2 字符串(指针)作函数参数.....	204
8.3.2 字符串.....	146	10.3.3 字符指针变量和字符数组 的区别.....	205
8.3.3 字符串数组.....	147	10.4 指针与结构体.....	206
8.4 重命名类型.....	149	10.4.1 指向结构体数据的指针变量.....	206
8.4.1 重命名类型的方法.....	149	10.4.2 指向结构体的指针作函数参数.....	207
8.4.2 重命名类型的作用.....	150	10.5 指针与链表.....	208
8.4.3 几点说明.....	150	10.5.1 单链表的数据描述.....	209
8.5 程序设计举例.....	150	10.5.2 单链表的建立.....	210
习题八.....	160	10.5.3 单链表的基本操作.....	212
第9章 结构体类型与共用体类型	162	10.5.4 双链表.....	213
9.1 结构体类型.....	162	10.6 指针与函数.....	214
9.1.1 结构体类型的定义.....	162	10.6.1 指向函数的指针变量.....	214
9.1.2 结构体变量的定义.....	164	10.6.2 指向函数的指针变量作 函数参数.....	215
9.1.3 结构体变量的引用.....	167	10.6.3 指针函数.....	216
9.1.4 结构体变量的初始化.....	167	10.7 指针作基类型.....	218
9.1.5 结构体作函数参数.....	170	10.7.1 指针数组.....	218
9.1.6 嵌套结构体.....	172	10.7.2 多级指针.....	219
9.2 结构体数组.....	173	10.7.3 指针数组作 main 函数的形参.....	220
9.3 共用体类型.....	175	10.8 程序设计举例.....	221
9.3.1 共用体类型定义.....	175	习题十.....	227
9.3.2 共用体变量的定义.....	176	第11章 文件类型	229
9.3.3 共用体变量的引用.....	176	11.1 文件类型与文件指针.....	229
9.4 枚举类型.....	178	11.1.1 文件概述.....	229
9.4.1 枚举类型的定义.....	178	11.1.2 文件类型.....	230
9.4.2 枚举类型变量的定义.....	179	11.1.3 文件指针变量.....	231
9.5 程序设计举例.....	181	11.2 文件的打开与关闭.....	231
习题九.....	187	11.2.1 文件的打开.....	231
第10章 指针类型	189	11.2.2 文件的关闭.....	233
10.1 指针与指针变量.....	189	11.3 文件的读写与建立.....	233
10.1.1 指针变量的定义.....	189	11.3.1 字符级数据的读和写.....	233
10.1.2 指针的运算.....	191		
10.1.3 利用指针处理简单数据.....	193		

11.3.2 字级数据的读和写	234	12.1.1 不带参数的宏定义	246
11.3.3 字符串级数据的读和写	234	12.1.2 带参数的宏定义	247
11.3.4 数据块级数据的读和写	235	12.1.3 预定义宏	248
11.3.5 格式化数据的读和写	235	12.1.4 取消宏定义	249
11.3.6 文件的建立步骤	236	12.2 文件包含	249
11.3.7 文件的读取控制	237	12.3 条件编译	249
11.4 文件辅助操作	238	12.4 图形处理	251
11.5 程序设计举例	239	12.4.1 图形处理流程	251
习题十一	245	12.4.2 图形处理函数	252
第 12 章 编译预处理和图形处理	246	习题十二	253
12.1 宏定义	246		

第三部分 C 语言程序设计提高篇

第 13 章 C 到 C++	257	13.4 类与对象	270
13.1 面向对象技术	257	13.4.1 类的定义	270
13.1.1 面向对象技术的由来和发展	257	13.4.2 对象的定义	272
13.1.2 面向对象技术的两大要素	257	13.4.3 构造函数和析构函数	273
13.1.3 面向对象技术的三大机制	258	13.4.4 继承性	274
13.1.4 面向对象程序设计	259	13.4.5 运行时的多态性	278
13.2 改进的 C 语言	259	13.5 Windows 编程基础	279
13.2.1 C++ 程序	259	13.5.1 Windows 程序	279
13.2.2 常规改进	261	13.5.2 Windows 程序结构	279
13.2.3 动态内存分配	263	13.5.3 一个简单的 VC++ 程序	280
13.2.4 引用	263	13.6 程序设计举例	283
13.2.5 函数	264	习题十三	286
13.2.6 重载	265	第 14 章 C 语言程序设计实例	287
13.3 C++ 的输入与输出	266	实例一 成绩处理程序	287
13.3.1 C++ 流类结构	267	实例二 电子词典程序	293
13.3.2 基本 I/O 操作	267	实例三 野人渡河程序	300
13.3.3 格式化 I/O	268		

附录 A C 语言中的关键字	309
附录 B 运算符和运算	310
附录 C 常用标准函数	311
参考文献	322

第一部分

C 语言程序设计入门篇

入门部分是整个 C 语言程序设计的基础篇，主要介绍：

- 基本语法成分
- C 语言程序的结构
- C 语言程序的实现
- 基本数据类型
- 数据描述方法
- 表达式与表达式语句
- 数据输入 / 输出
- 顺序结构程序设计
- 选择结构程序设计
- 循环结构程序设计
- 函数的运用
- 程序设计步骤
- 结构化程序设计方法初步

通过这部分的介绍，读者应重点掌握 C 语言的基本语法以及 C 语言程序的结构、数据与操作的基本描述方法，了解结构化程序设计方法，并尝试编制简单的 C 语言程序。

第1章 C语言基础

程序设计语言是程序设计人员与计算机进行对话的语言，它遵循一定的规则和形式，构成程序和软件的开发平台。为满足计算机的各种应用，人们设计了上百种程序设计语言。

C语言是一种功能通用、应用广泛、很有发展前途的高级语言，既可用于开发系统程序，又可用于开发各种应用程序。

本章作为开篇，将介绍C语言的特点、C语言的基本语法成分、C语言程序的结构及C语言程序的实现。

1.1 C语言概述

1.1.1 C语言的产生与发展

1. C语言的产生

1972年~1973年间，C语言诞生于美国的贝尔实验室，由D.M.Ritchie创建。

C语言的产生基于两个方面的需要。一是为满足UNIX操作系统开发的需要。UNIX操作系统是一个通用的、复杂的计算机管理系统。二是为拉近高级语言与硬件之间距离的需要。C语言集高级语言与汇编语言的优点于一身。

C语言面对实际应用的需要而产生，直至今日仍不改初衷。

2. C语言的发展

对C语言追本求源，可追溯到ALGOL语言。1960年出现的ALGOL语言是一种面向问题的高级语言，远离硬件，不适于开发系统软件。1963年，英国剑桥大学推出CPL语言，CPL语言比ALGOL语言更接近硬件一些，但规模较大，难以实现。1969年，剑桥大学的Martin Richards对CPL语言进行简化，推出BCPL语言。1970年，贝尔实验室的Ken Thompson为DEC公司的PDP-7计算机上运行的一种早期UNIX操作系统设计了一种类BCPL语言，称为B语言。B语言规模小，接近硬件，1971年在PDP-11计算机上实现。BCPL语言和B语言都是无类型的语言，过于简单，功能有限。

1972年~1973年间，贝尔实验室的D.M.Ritchie在保留B语言优点的基础上，创建了C语言。1973年，UNIX操作系统被用C语言改写，称为UNIX第五版。最初的C语言只是一种UNIX操作系统的工作语言，依附于UNIX系统，主要在贝尔实验室内部使用。UNIX以后的第六版、第七版、SYSTEM III和SYSTEM V都是在第五版的基础上发展起来



的。这期间 C 语言也被多次改进。1975 年, UNIX 的第六版公布, C 语言开始受到人们的普遍关注。

UNIX 操作系统的广泛使用, 促进了 C 语言的迅速发展及普及, C 语言的发展与普及反过来又促进了 UNIX 操作系统的推广。1978 年, 出现了独立于 UNIX 和 PDP 计算机的 C 语言, 从此 C 语言被迅速移植到大、中、小型与微型机上。当年, B.W.Kernighan 和 D.M.Ritchie 以 UNIX 第七版的 C 编译程序为基础, 出版了影响深远的名著《The C Programming Language》。

C++语言是 C 语言发展的新阶段, 是一种更好的 C 语言, 是应用广泛的面向对象的程序设计语言。

目前, C 语言已经发展成为迄今为止最流行的计算机程序设计语言。

3. C 语言的标准与方言

Kernighan 和 Ritchie 的名著《The C Programming Language》出版不久, 各种 C 语言编译系统纷纷出现。由于这本书的存在, 使得绝大多数 C 语言保持着高度的兼容性。这本书给出了 C 语言的经典定义, 它所描述的 C 语言成为 C 语言世界的当然标准, 简称 K&R C。

然而, C 语言的广泛应用还是在不同的开发团体之间出现了问题, 各机构推出了自己的 C 语言版本, 某些执行过程的微小差别却不时引起 C 程序之间的不兼容。美国国家标准协会(ANSI)从 1983 年开始经过长达五年的努力, 制定了 C 语言的新标准——ANSI C。现在提及 C 语言的标准都是指新标准。ANSI C 比原标准 C 有很大的发展, 解决了经典定义中的二义性, 给出了 C 语言的新特点。Kernighan 和 Ritchie 也以新标准改写了他们的经典著作。任何 C 程序都必须遵循 ANSI C 标准, 本教材的主体也以 ANSI C 作为基础。

目前, 微机上广泛使用的 C 语言为 Borland 公司的产品 Turbo C 和 Microsoft 公司的产品 MS C。Turbo C 2.0 完全支持新标准 C, 并提供集成开发操作环境。本教材第一、二部分以 Turbo C 2.0 作为实践环境。C++主要有 Borland C++和 Visual C++, 本教材第三部分选择 Visual C++ 6.0 作为实践环境。

1.1.2 C 语言的特点

C 语言是一种通用、灵活、结构化、标准化、使用普遍的编程语言, 能完成用户想得到的任何任务, 特别适合于进行系统程序设计和对硬件进行操作的场合。

C 语言本身不对程序员施加过多限制, 因而成为专业程序员优先选择的语言。

C 语言的主要特点如下:

(1) C 语言简洁、紧凑, 压缩了一切不必要的成分。

ANSI C 有 32 个关键字, Turbo C 有 58 个关键字, 9 种控制语句, 书写形式自由。

(2) C 语言运算丰富, 将括号、赋值、强制类型转换、取变量地址等都以运算实现。

ANSI C 提供 34 种运算, Turbo C 提供 44 种运算, 灵活使用这些运算可以实现其它高级语言难以实现的操作。C 语言的表达式简练、多样、灵活、实用, C 语言的表达式加上分号就可以构成语句。

(3) C 语言数据类型丰富, 具有现代语言的各种数据类型。同时, 用户还能扩充数据类型, 能够实现各种复杂的数据结构, 完成各种问题的数据描述。



尤其是 C 语言的指针类型非常有特色,可指向各种数据,完成对各种数据的高效处理。C 语言对数据不但作类型上的描述,还提供存储属性考虑。

(4) C 语言是一种结构化程序设计语言,具有结构化语言所要求的基本结构。

C 语言用函数作为结构化程序设计的实现工具,实现程序的模块化。C 程序由若干程序文件组成,一个程序文件由若干函数构成。

(5) C 语言是高级语言中的低级语言。C 语言允许直接访问物理地址,能进行位运算,能实现汇编语言的大部分功能,能直接对硬件进行操作。

(6) C 语言提供编译预处理机制,有利于大程序的编写和调试。

(7) C 语言编译系统小,生成的目标代码质量高,程序的执行效率高。

(8) C 语言的输入和输出功能用库函数实现,编写的程序移植性好。

(9) C 语言语法限制不太严格,程序设计自由度大,对程序员要求较高。

以上是关于 C 语言的一些一般特点,要真正了解 C 语言的特点,必须等学完 C 语言以后。但有一点可以确信,C 语言是程序员的语言,C 语言提供了程序员想要的一切。

1.2 基本语法成分

本节介绍 C 语言的字符集、关键字及标识符等基本语法成分。

1.2.1 字符集

字符是可以区分的最小符号,字符集构成语言与程序的原始基础。C 语言的字符集是 ASCII 字符集的一个子集,包括英文字母、数字及特殊字符。

- 英文字母: a~z 和 A~Z。
- 数字: 0~9。
- 特殊字符: 空格、!、"、#、\$、%、^、&、*、-、_、+、=、~、<、>、/、\、|、.、:、;、?、'、"、(、)、[、]、{、}。

由字符集中的字符可以构成 C 语言进一步的语法成分,如标识符、关键字、特殊的运算符等。

1.2.2 标识符

标识符在语句与程序中用来标识各种语句与程序成分,用来命名语句与程序中的一些实体,如变量、常量、函数、类型、标号等的名字。

C 语言规定,标识符必须以字母或下划线开头,是字母、数字、下划线的序列。

以下标识符是合法的标识符:

```
i, j, k, x, c, a1, a2, op, y_1, zhou_prg, radius, prime, program, sort,
max, min, prg_1, cout, sun, day
```

以下是不合法的标识符:



a.1, lcomputer, x+y, !abc, 99999, \$100, π , 3c

C 语言中的标识符严格区分字母的大小写, 标识符 abc 与标识符 ABC 是不同的标识符。习惯上, 符号常量用大写字母表示, 变量名称用小写字母表示。

标准 C 中, 标识符的长度可以任意, 一般有效长度为 8 个字符, Turbo C 中标识符最大长度为 32 个字符。

标识符取名时不能与关键字同名, 也不要与系统预先定义的标准标识符(如标准函数)同名。系统使用的一些内部标识符往往以下划线开头, 为避免与系统使用的内部标识符发生冲突, 用户定义的标识符一般也不要以下划线开头。

标识符的名称一般选用相应的英文单词或拼音形式缩写, 尽量不要使用简单代数符号, 如 a、b、c、x、y、z 等。最好能“见名知义”, 以提高程序的可读性。标识符大多数采用“匈牙利”表示法, 即每个单词第一个字母大写, 如 StudentName、YearMonthDate。

由系统预先定义的标识符称为标准标识符, 由用户定义的标识符称为自定义标识符。程序设计中往往需根据实际需要定义大量的标识符。标识符必须先定义, 后使用。

1.2.3 关键字

关键字又称为保留字, 由系统提供, 用以表示特定的语法成分, 是构成 C 语言的语法基础。实际上关键字是单列的一些非常特殊的有特定含义的准标识符。如表示标准类型的名称 int、float 等关键字, 定义类型的 array、struct 等关键字, 表示语句特征的 if、switch、for、while 等关键字, 表示运算符的 sizeof 等关键字。

常见关键字有 32 个, 参见附录 A。这些关键字将逐步引入, 但希望读者能提前予以特别关注。

关键字有特定的语法含义, 绝对不允许用户重新定义。关键字在程序中像国家的名字一样, 也绝对不能拼错。

1.2.4 运算符

运算符实际上可以认为是系统定义的函数名字, 这些函数作用于运算对象, 得到一个结果值。运算符通常由 1 个或多个字符构成。C 语言沿用了大量的常规运算符, 如加法运算符“+”、减法运算符“-”、地址运算符“&”、大于运算符“>”、不等运算符“!=”、逻辑与运算符“&&”、条件运算符“?:”、点运算符“.”、字节数运算符“sizeof”等。

根据运算对象的个数不同, 可分为单目运算符、双目运算符和三目运算符, 又称为一元运算符、二元运算符和三元运算符。

C 语言的运算符非常丰富, 这些将在介绍数据类型与表达式的章节中予以详述, 并请读者参考附录 B。

1.3 C 语言程序的结构

了解并掌握 C 语言程序的结构, 对于 C 语言程序的设计至关重要。在讨论 C 语言程



序结构之前, 先看几个例子, 以取得对C语言程序结构的一些感性认识。

例 1-1 在屏幕上输出“C语言程序设计”。

```
/*程序 1-1, 输出“C语言程序设计”*/  
#include "stdio.h"  
  
main()  
{printf("C语言程序设计");  
}
```

运行结果: C语言程序设计

例 1-2 求两个整数之和。

```
/*程序 1-2, 求两个整数之和*/  
#include "stdio.h"  
  
main()  
{ int a, b;  
  int sum;  
  printf("请输入两个整数 a, b: ");  
  scanf("%d, %d", &a, &b);  
  sum=a+b;  
  printf("和=%d\n", sum);  
}
```

输入数据: 1999, 5

运行结果: 和=2004

例 1-3 求两个数的最大值。

```
/*程序 1-3, 求两个数的最大值*/  
#include "stdio.h"  
  
float max(x, y)          /*求最大值函数*/  
float x, y;  
{ float t;  
  if (x>y) t=x;  
  else t=y;  
  return(t);  
}  
  
main()                  /*主函数*/  
{ float a, b;  
  float m;  
  printf("请输入两个数: ");  
  scanf("%f, %f", &a,&b);  
  m=max(a, b);          /*调用求最大值的函数*/  
  printf("最大值=%6.2f\n", m);  
}
```

输入数据: 12.1, 3.45