

C How To Program

C语言 大学教程

本书注重教材的可读性和适用性，在每章开头都给出了学习意义和学习目标；对关键知识点进行了详细的说明，并附有大量的图表使读者能正确、直观地理解问题；示例程序由浅入深，强化知识点、算法、编程方法与技巧，并给出了详细的解释；为了帮助初学者正确地掌握C语言的语法特点，每章还列举出了初学者在编程过程中易犯的错误。



冯博琴

西安交通大学教授，计算机教学实验中心主任，首届国家级教学名师，现任教育部计算机基础课程教学指导委员会副主任委员，全国计算机教育研究会副会长。从事计算机教学和科研工作40多年，成果颇丰，提出的“精讲多练、教考分离、机试为主”教学方法在国内产生了较大影响。编写了多部国家级和省部级规划教材，主编出版的教材已逾30部，译著20余部。获国家级教学成果一等奖2项、二等奖3项，国家级优秀教材一、二等奖各一项，国家精品课程两门；承担国家十五规划教材、教育部多项国家级教改项目。先后获得首届国家级教学名师奖、宝钢教育基金优秀教师特等奖、全国模范教师、全国五一劳动奖章、全国师德先进个人称号，享受政府特殊津贴。是国家级计算机实验教学示范中心、国家级教学团队牵头人。

本教材的结构框图



免费提供

PPT等教学相关资料



人民邮电出版社
教学服务与资源网
www.ptpedu.com.cn

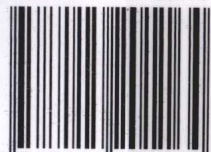
教材服务热线：010-67170985

反馈/投稿/推荐信箱：315@ptpress.com.cn

人民邮电出版社教学服务与资源网：www.ptpedu.com.cn



ISBN 978-7-115-20116-4



9 787115 201164 >

ISBN 978-7-115-20116-4

定价：32.00 元

封面设计：董志桢

人民邮电出版社网址：www.ptpress.com.cn

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 大学教程

C How To Program

王浩鸣 郭晔 主编

冯博琴 主审



名家系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C语言大学教程 / 王浩鸣, 郭晔主编. — 北京: 人民邮电出版社, 2009. 10

21世纪高等学校计算机规划教材
ISBN 978-7-115-20116-4

I. ①C… II. ①王… ②郭… III. ①
C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(2009)第157623号

内 容 提 要

本书对 C 语言的基本构成、语法规则及 C 程序的编辑、编译和执行过程, 从一般程序设计语言的共性到 C 语言自身的特性, 从 C 语言的语法规则到其内部实现, 做了系统的介绍。具体包括 C 语言的基本数据类型和表达式、算法与 C 语言程序的关系、分支结构和循环结构程序设计、函数、数组、指针、结构体、文件等内容, 并通过实验和综合开发示例, 较为全面地介绍了 C 语言程序设计的步骤和方法。为了拓展学生的知识结构, 在本书的最后对 C++进行了简要概述。

本书各章均配有多媒体电子教案和习题, 以方便教师教学和学生自学。为了帮助学生加深对教材内容的理解, 培养学生的动手能力, 书后还特别编写了配套实验。

本书可作为高等院校各专业学生程序设计课程的教材, 也可作为相关人员的自学参考书。

21 世纪高等学校计算机规划教材

C 语言大学教程

-
- ◆ 主 编 王浩鸣 郭 晔
主 审 冯博琴
责任编辑 刘 博
 - ◆ 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>
北京铭成印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 19
字数: 497 千字 2009 年 10 月第 1 版
印数: 1—3 000 册 2009 年 10 月北京第 1 次印刷

ISBN 978-7-115-20116-4

定价: 32.00 元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223
反盗版热线: (010) 67171154

出版者的话

计算机科学与技术日新月异的发展,对我国高校计算机人才的培养提出了更高的要求。许多高校主动研究和调整学科内部结构、人才培养目标,提高学科水平和教学质量,精炼教学内容,拓宽专业基础,优化课程结构,改进教学方法,逐步形成了“基础课程精深,专业课程宽新”的良性格局。作为大学计算机教材建设的生力军,人民邮电出版社始终坚持服务高校教学、致力教育资源建设的出版理念,在总结前期教材建设的成功经验的同时,深入调研和分析课程体系,并充分结合我国高校计算机教育现状和改革成果,推出“推介名师好书,共享教育资源”的教材建设项目,出版了“21世纪高等学校计算机规划教材”名家系列。

本套教材的突出特点如下:

(1) 作者权威 本套教材的作者均为国内计算机学科中的学术泰斗或高校教学一线的教学名师,他们有着深厚的科研功底和丰富的教学经验。可以说,这套教材汇聚了众师之精华,充分显示了这套教材的格调和品位。无论是刚入杏坛的年轻教师,还是象牙塔内的莘莘学子,细细品读其中的章节文字,定会受益匪浅。

(2) 定位准确 本套教材是为普通高等院校的学生量身定做的精品教材。具体体现在:一是本套教材的作者长期从事一线科研和教学工作,对高校教学有着深刻而独到的见解;二是本套教材在选题策划阶段便多次召开调研会,对普通高校的教学需求和教材建设情况进行充分摸底,从而保证教材在内容组织和结构安排上更加贴近实际教学;三是组织有关作者到较为典型的普通高等院校讲授课程教学方法,深入了解教师的教学需求,充分把握学生的理解能力,以教材内容引导授课教师严格按照科学方法实施教学。

(3) 教材内容与时俱进 本套教材在充分吸收国内外最新计算机教学理念和教育体系的同时,更加注重基础理论、基本知识和基本技能的培养,集思想性、科学性、启发性、先进性和适应性于一身。

(4) 一纲多本,合理配套 根据不同的教学法,同一门课程可以有多本不同的教材,教材内容各具特色,实现教材系列资源配套。

总之,本套教材中的每一本精品教材都切实体现了各位教学名师的教学水平,充分折射出名师的教学思想,淋漓尽致地表达着名师的教学风格。我们相信,这套教材的出版发行一定能够启发年轻教师们真正领悟教学精髓,教会学生科学地掌握计算机专业的基本理论和知识,并通过实践深化对理论的理解,学以致用。

我们相信,这套教材的策划和出版,无论在形式上还是在内容上都能够显著提高我国高校计算机专业教材的整体水平,为培养符合时代发展要求的具有较强国际竞争力的高素质创新型计算机人才,为我国普通高等教育的计算机教材建设工作做出新的贡献。欢迎各位老师和读者给我们的工作提出宝贵意见。

前言

C 语言是目前最为流行的通用程序设计语言之一,是许多计算机专业人员和计算机爱好者学习程序设计语言的首选,因此,编写一本适合于初学者的 C 语言教材具有重要的意义。

本书注重教材的可读性和适用性,在每章开头都给出了学习意义和学习目标;对关键知识点进行了详细的说明,并附有大量的图表使读者能正确、直观地理解问题;示例程序由浅入深,强化知识点、算法、编程方法与技巧,并给出了详细的解释;为了帮助初学者正确地掌握 C 语言的语法特点,每章还列举出了初学者在编程过程中易犯的错误。

本书的作者都是长期在高校从事计算机软件教学的一线教师,有丰富的教学经验和科研开发能力。本书在内容的选择上力求概念清晰、重点突出、案例新颖;内容组织上由浅入深、循序渐进;编写风格上语言准确、言简意赅。另外,本书理论、实践并重,结构安排合理,突出了 C 语言的特点。

本书共分 12 章。第 1 章用 C 语言编写程序,介绍了 C 语言程序设计结构和书写风格及 C 语言程序的编译和执行;第 2 章基本数据类型和表达式,介绍了 C 语言的基本数据类型、不同类型变量赋值的方法及运算符构成的表达式;第 3 章算法与 C 语言程序,介绍了计算机求解问题的步骤、算法作用及算法的构造;第 4 章分支结构程序设计,介绍了程序控制流程及分支结构程序设计的方法;第 5 章循环结构程序设计,介绍了循环结构构成及利用循环语句进行程序设计的方法;第 6 章函数,介绍了函数使用方式及使用技巧;第 7 章数组,介绍了数组的作用及正确使用数组的重要性;第 8 章指针,介绍了指针的概念并通过具体的例子重点介绍了指针的使用方法;第 9 章结构体,介绍了利用结构体处理大量有规律数据的方法;第 10 章文件,介绍了文件的概念及利用文件实现数据存储的具体实例;第 11 章 C 语言程序设计方法,介绍了结构化程序设计的基本方法,程序调试过程中可能出现的问题以及解决问题的思路;第 12 章 C++概述,简要介绍了 C++程序设计语言的特点以及使用 C++程序设计语言进行程序编制的方法。另外,本书为任课老师免费提供电子课件,其中包括全部例题和习题的源程序文件,可到人民邮电出版社教学服务与资源网(www.ptpedu.com.cn)上免费下载。

本书由王浩鸣、郭晔担任主编。第 1 章、第 12 章由王浩鸣编写,第 2 章、第 3 章由魏晋雁编写,第 4 章、第 5 章由李翠编写,第 6 章由郭晔编写,第 7 章、第 10 章由史西兵编写,第 8 章、第 11 章由李建廷编写,第 9 章由黄艳编写。全书由王浩鸣、郭晔统稿。

本书适用于高等学校计算机专业的本科生、研究生使用,也可作为各专业公共教材和全国计算机等级考试参考书。

在本书的编写过程中,承蒙冯博琴教授亲自指导,作者所在学院老师对

书中内容提出了许多宝贵意见，对他们给予的支持和帮助，在此一并表示衷心感谢。

由于水平有限，书中难免存在疏漏之处，恳请广大读者批评指正。

作 者

2009 年 8 月于西安

目 录

第 1 章 用 C 语言编写程序.....1

1.1 计算机语言与程序设计语言.....1
1.1.1 程序设计语言的分类.....2
1.1.2 C 语言简介.....3
1.1.3 C 语言的优点.....3
1.1.4 C 语言的缺点.....4
1.2 C 语言程序的结构.....4
1.3 C 语言的基本输入输出函数.....6
1.3.1 格式化输入输出函数.....6
1.3.2 非格式化输入/输出函数.....10
1.4 开发 C 语言程序的步骤.....12
小结.....13
习题.....13
实验.....13

第 2 章 基本数据类型和表达式.....17

2.1 数据类型.....17
2.2 常量和变量.....18
2.2.1 常量.....18
2.2.2 变量.....22
2.2.3 常用的输入/输出库函数.....26
2.3 运算符与表达式.....32
2.3.1 算术运算符和算术表达式.....33
2.3.2 赋值运算符与赋值表达式.....33
2.3.3 关系运算符与关系表达式.....34
2.3.4 逻辑运算符与逻辑表达式.....35
2.3.5 问号表达式与逗号表达式.....37
2.3.6 自增与自减运算符.....39
2.4 数据的存储和类型转换.....39
2.4.1 数据的存储.....39
2.4.2 数据类型转换.....41
小结.....42
习题.....43

第 3 章 算法与 C 语言程序.....45

3.1 计算机求解问题的步骤.....45

3.2 算法与程序.....46
3.2.1 算法的特征.....46
3.2.2 算法的实现.....47
3.2.3 算法与程序结构.....47
3.3 算法的描述.....48
3.3.1 用自然语言表示算法.....48
3.3.2 用流程图表示算法.....48
3.3.3 用伪代码表示算法.....51
3.4 C 语句分类.....51
小结.....52
习题.....53

第 4 章 分支结构程序设计.....54

4.1 关系表达式和逻辑表达式.....54
4.1.1 关系表达式.....54
4.1.2 逻辑表达式.....55
4.2 if 语句.....59
4.2.1 基本的 if 语句.....59
4.2.2 嵌套的 if 语句.....62
4.2.3 条件运算符与条件表达式.....67
4.3 switch 语句.....68
4.3.1 switch 语句的一般形式.....68
4.3.2 switch 语句的使用说明.....70
小结.....72
习题.....72
实验.....74

第 5 章 循环结构程序设计.....78

5.1 C 语言的循环语句.....78
5.1.1 while 语句.....78
5.1.2 do-while 语句.....80
5.1.3 for 语句.....82
5.1.4 for 语句的形式变化.....84
5.2 break 语句和 continue 语句.....85
5.2.1 break 语句.....85
5.2.2 continue 语句.....86
5.3 循环嵌套.....87

5.4 循环程序设计	89	7.3.4 二维数组应用举例	130
小结	92	7.4 字符数组	132
习题	92	7.4.1 字符数组的定义	132
实验	94	7.4.2 字符数组的初始化	132
第 6 章 函数	97	7.4.3 字符数组的引用	133
6.1 函数定义	97	7.4.4 字符数组的输入输出	134
6.1.1 函数概念	97	7.4.5 字符串处理函数	135
6.1.2 函数定义	98	7.4.6 字符数组应用举例	137
6.2 函数调用	99	小结	139
6.2.1 函数调用过程	99	习题	139
6.2.2 函数调用形式	99	实验	140
6.2.3 函数的形式参数和实际参数	100	第 8 章 指针	143
6.2.4 参数传递方式	101	8.1 指针概述	143
6.2.5 函数结果返回	102	8.1.1 指针变量的定义	143
6.2.6 函数的嵌套调用	103	8.1.2 指针变量的引用	146
6.2.7 函数的声明	104	8.1.3 指针的基本运算	148
6.3 递归函数	105	8.1.4 指针作为函数的参数	151
6.3.1 递归函数基本概念	105	8.2 指针和数组	154
6.3.2 递归程序设计	107	8.2.1 数组的指针变量	155
6.4 变量与函数	109	8.2.2 数组元素的引用	155
6.4.1 局部变量和全局变量	109	8.2.3 数组名作为函数参数	158
6.4.2 变量生命周期和静态局部变量	112	8.2.4 指向多维数组的指针和指针变量	161
6.4.3 寄存器变量和外部变量	113	8.2.5 动态数组的实现	166
6.5 编译预处理	114	8.3 指针和字符串	168
6.5.1 宏定义	114	8.3.1 字符串的表示和引用	168
6.5.2 条件编译	119	8.3.2 常用的字符串处理函数	170
小结	121	8.3.3 字符串指针作为函数参数	172
习题	122	8.3.4 字符数组与字符串指针的区别	173
实验	122	8.4 指针和函数	175
第 7 章 数组	124	8.4.1 指向函数的指针	175
7.1 概述	124	8.4.2 返回指针值的函数	178
7.2 一维数组	124	8.5 指针数组和指向指针的指针	180
7.2.1 一维数组的定义	125	8.5.1 指针数组	180
7.2.2 一维数组的初始化	125	8.5.2 指向指针的指针	183
7.2.3 一维数组的引用	126	8.5.3 带参数的 main 函数	186
7.2.4 一维数组的应用举例	127	小结	188
7.3 二维数组	129	习题	190
7.3.1 二维数组的定义	129	实验	193
7.3.2 二维数组的初始化	130	第 9 章 结构体	197
7.3.3 二维数组的引用	130	9.1 结构体概念	197

9.1.1 结构体类型的定义	197	小结	244
9.1.2 结构体变量的定义和引用	198	习题	244
9.2 结构体数组	201	实验	246
9.2.1 结构体数组的定义和引用	201	第 11 章 C 语言程序设计方法	248
9.2.2 结构体数组的初始化	202	11.1 结构化程序设计方法	248
9.3 结构体指针	204	11.1.1 程序设计方法的发展	248
9.3.1 结构体指针变量的概念和使用	204	11.1.2 “自顶而下，逐步细化”的 设计过程	249
9.3.2 结构体指针作为函数的参数	206	11.1.3 模块化程序设计	251
9.4 链表	207	11.1.4 结构化程序设计	252
9.4.1 链表的定义	207	11.2 程序设计风格	253
9.4.2 链表的常用操作	208	11.2.1 源程序文档化	254
9.5 联合	216	11.2.2 语句结构标准化	256
9.5.1 联合的定义	216	11.3 程序设计中要注意的问题	261
9.5.2 联合变量的定义和引用	217	11.3.1 程序设计的基本模式	261
9.6 枚举	219	11.3.2 程序编制中的常见错误	265
9.6.1 枚举类型的定义	219	11.3.3 程序调试中的常见错误	269
9.6.2 枚举变量的定义和引用	219	小结	271
9.7 自定义类型	220	习题	272
9.8 位运算与位段	221	实验	272
9.8.1 位运算和位运算符	221	第 12 章 C++概述	277
9.8.2 位段	223	12.1 C++语言的起源与发展	277
小结	224	12.2 面向对象的程序设计方法	278
习题	225	12.2.1 概述	278
实验	225	12.2.2 基本概念	278
第 10 章 文件	227	12.2.3 C++的特色	281
10.1 文件概述	227	12.3 标准模板库	282
10.2 文件的打开与关闭	228	12.3.1 标准模板库简介	282
10.2.1 文件的打开	228	12.3.2 标准模板库和 C++	283
10.2.2 文件的关闭	230	12.3.3 标准模板库结构	284
10.3 文件的读写	230	12.3.4 标准模板库和 C++标准函数库	285
10.3.1 字符输入输出函数	230	12.4 C++程序示例	286
10.3.2 字符串输入/输出函数	233	12.4.1 问题分析	287
10.3.3 数据块输入/输出函数	235	12.4.2 哈希表简介	287
10.3.4 格式化输入/输出函数	236	12.4.3 程序实现	288
10.4 文件的定位	239	小结	292
10.4.1 指针指向文件开头的函数	239	参考文献	293
10.4.2 设置文件指针位置函数	240		
10.4.3 获得文件当前读写位置函数	242		
10.5 文件状态检测	242		
10.5.1 文件读写结束标志函数	243		
10.5.2 检查文件出错函数	244		

第 1 章

用 C 语言编写程序

【教学目的】

- 了解 C 语言程序的结构和书写风格；
- 了解 C 语言程序的编译和执行方法。

随着计算机硬件技术的飞速发展，包括 CPU、内存、硬盘在内的硬件运行速度得到了极大的提高，而且单位价格也呈下降趋势，程序员可以利用更为先进的程序设计方法与硬件平台进行应用软件的开发工作。C 语言作为介于汇编语言和高级语言之间的语言，是集汇编语言和高级语言的优点于一身的程序设计语言，掌握用 C 语言开发应用程序的方法具有重要的意义。

1.1 计算机语言与程序设计语言

计算机语言（Computer Language）是用于人与计算机之间通信的语言，是人与计算机之间传递信息的媒介，目前人对计算机发出的命令几乎都是通过计算机语言进行的。

程序设计语言（Programming Language）是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的行动。

程序设计语言原本是被设计成专门使用在计算机上的，但它们也可以用来定义算法或者数据结构。正是因为如此，程序员才可能使程序代码更容易阅读。

程序设计语言往往使程序员能够比使用机器语言更准确地表达他们所想表达的目的。对那些从事计算机科学的人来说，懂得程序设计语言是十分重要的，因为目前几乎所有的计算都需要程序设计语言才能完成。

在过去的几十年间，大量的程序设计语言被发明、取代、修改或组合在一起。尽管人们多次试图创造一种通用的程序设计语言，却没有一次尝试是成功的，主要原因是编写程序的初衷各不相同，程序编制者之间技术水平差距很大，不同程序之间的运行成本（Runtime Cost）各不相同，所有这些都使发明一种通用程序设计语言的想法难以实现。

采用某种程序设计语言设计的程序需要被翻译成计算机语言后才能被计算机所识别并执行，这个过程称为编译（Compile）或解释（Interpret）。

所谓编译是指将所要翻译的采用高级语言编写的程序代码作为一个整体进行翻译，生成可以

被计算机所识别的二进制代码。进行编译操作的程序块称为编译器。因此，编译器将人可阅读的程序文本（称为源代码）作为输入，输出计算机可识别的可执行文件（Object Code）。该可执行文件是机器语言，可由计算机的 CPU 直接运行。

所谓解释是指采用高级语言编制的程序代码在运行时才被即时翻译，运用这种方法翻译的程序运行速度往往比编译的程序慢，但更具灵活性，它们能够与执行环境互相作用，使程序的编制者可以通过观察程序运行情况更好地了解程序的运行过程。

计算机语言的概念比程序设计语言要更为广泛。例如 HTML 是一种标记性语言，也属于计算机语言，但并不是程序设计语言。

1.1.1 程序设计语言的分类

程序设计语言按照语言级别可以分为低级程序设计语言（Low-Level Programming Language）和高级程序设计语言（High-Level Programming Language）。这里的高级与低级是指编写程序的指令是否更为抽象化或与人类的思维更为接近。

低级语言一般包括机器语言和汇编语言。机器语言是指采用 0、1 代码表示操作码与操作数的一种程序设计方式，利用机器语言编写的程序，其每条指令可以直接被 CPU 执行，不需要经过转换过程，因此，其优点是指令与特定的机器相关，执行效率最高；缺点是使用繁琐复杂，编写费时，易出差错，而且对于不同类型的 CPU，其指令各不相同，同一个程序不能在不同的 CPU 硬件平台上运行，可移植性差。

汇编语言是在机器语言的基础上，将操作码采用一定的符号来表示，比机器语言易于理解与记忆，但在 CPU 执行之前需要先转换为机器能够识别的二进制代码。其优缺点与机器语言相同。

高级语言是相对于低级语言而言的，它以人类易于接受的文字来表达解决问题的全过程。用户不必费时费力去记忆难以理解的 0、1 代码，而是可以把主要精力集中于解决问题的算法上。因此，高级语言的优点是编写简单，可读性高。因为高级语言本身不依赖于特定的 CPU，所以用高级语言编写的程序稍加修改就可以在不同的 CPU 硬件平台上运行，可移植性较好。

与汇编语言类似，使用高级语言编写的程序需要经过编译或解释先变成机器能够识别的二进制代码，才能在机器上正确执行。

程序设计语言可以按照不同的分类方法进行分类。

（1）按照应用范围可分为通用语言与专用语言。用户根据解决问题的算法，利用高级语言提供的命令进行程序设计，被解决的问题可以是各种各样的，与行业、使用范围没有关系，这类语言称为通用语言，如 FORTRAN、PASCAL、C、Visual Studio 系列等。与之相对应，如果该语言只能使用在特定的环境，则称为专用语言，如 APT 语言是一种专用于控制数控机床的各个部分进行协调工作的一种语言。

（2）按照使用方式可分为交互式语言和非交互式语言。具有反映人机交互作用的语言成分的语言称为交互式语言，如 BASIC 等。不反映人机交互作用的语言称为非交互式语言，如 FORTRAN、PASCAL、C 等都是非交互式语言。

（3）按用户的要求可分为过程式语言和非过程式语言。过程式语言的主要特征是，用户按自己的意愿指明一系列运算执行的顺序，以表示相应的计算过程，如 FORTRAN、PASCAL、C 语言等，但这些顺序未必是最合理的。非过程式语言是指由事件驱动型的，规定当程序中某个事件发生时，就执行相应的处理过程，这是目前流行的面向对象程序设计方法的根本思想，如 Visual

C++等。

1.1.2 C 语言简介

C 语言是一种通用的程序设计语言，它主要用来进行系统程序设计，具有高效、灵活、功能丰富、表达力强和移植性好等特点，在程序员中备受青睐。

C 语言是由 UNIX 的研制者丹尼斯·里奇 (Dennis.Ritchie) 和肯·汤普逊 (Ken.Thompson) 于 1970 年研制出的 B 语言的基础上发展和完善起来的。C 语言可以广泛应用于不同的操作系统中，如 UNIX、MS-DOS、Microsoft Windows 及 Linux 等。C 语言是一种面向过程的程序设计语言，同时具有高级语言和汇编语言的优点。

为了使 UNIX 操作系统得到推广，1977 年，Dennis.Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》，即著名的 ANSI C。

1978 年，Brian W. Kernighian 和 Dennis.Ritchie 出版了著作《C 语言程序》(The C Programming Language)，从而使 C 语言成为当时世界上流行最广泛的高级程序设计语言。

1988 年，随着微型计算机的日益普及，C 语言出现了许多版本。由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准组织 (ANSI) 为 C 语言制定了一套 ANSI 标准，成为现行的 C 语言标准。

C 语言发展迅速，而且成为最受欢迎的语言之一，主要因为它具有强大的功能。许多著名的系统软件，如 dBase III PLUS、dBase IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序，就更能显示 C 语言的优势。

1.1.3 C 语言的优点

(1) 简洁紧凑、灵活方便。C 语言一共只有 32 个关键字，9 种控制语句，程序书写自由，主要用小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

(2) 运算符丰富。C 语言的运算符包含的范围很广泛，共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(3) 数据结构丰富。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。它们能用来实现各种复杂的数据类型的运算。

(4) C 语言是结构式语言。结构式语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护以及调试。C 语言是以函数形式提供给用户的，这些函数可方便地调用，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

(5) C 语言语法限制不太严格，程序设计自由度大。虽然 C 语言也是强类型语言，但它的语法比较灵活，允许程序编写者有较大的自由度。

(6) C 语言允许直接访问物理地址，可以直接对硬件进行操作。因此 C 语言既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元，可以用来编写系统软件。

(7) C 语言程序生成代码质量高，程序执行效率高，一般只比汇编程序生成的目标代码效率低 10%~20%。

(8) C 语言适用范围大,可移植性好。C 语言有一个突出的优点就是适合于多种操作系统,如 DOS、UNIX,也适用于多种机型。

1.1.4 C 语言的缺点

(1) 数据的封装性不好。这点使得 C 语言在数据的安全性上有很大缺陷,这也是 C 语言和 C++ 的一大区别。

(2) 语法限制不太严格,对变量的类型约束不严格,影响程序的安全性,对数组下标越界不作检查等。从应用的角度,C 语言比其他高级语言较难掌握。

(3) 指针可能被滥用。指针是 C 语言的一大特色,C 语言优于其他高级语言的一个重要原因就是因为它有指针操作,可以直接进行底层的操作。但同时 C 语言的指针操作也给它带来了许多不安全的因素,可能被滥用。而 C++ 在这方面做了很好的改进,在保留了指针操作的同时又增强了安全性。

1.2 C 语言程序的结构

通过下面的例子可以说明 C 语言程序的结构:

```
/* 文件名: Ex01_01.C */           // 程序说明部分
/* 功能: 输出指定的文字行 */      // 程序说明部分
#include <stdio.h>                 // 注明使用的库函数
int main(void)                     // 主函数从 main() 开始
{
    printf("Hello, World!\n");     // 从标准输出设备输出字符串
    return 0;                      // 返回
}
```

该程序的运行结果如下:

```
Hello World!
```

在上面的程序中,包括以下几个部分。

(1) 程序说明部分。用来说明该程序的文件名、程序的功能等信息,也称为注释行。该部分内容起备忘的作用,它们并不会被编译到最终的可执行文件中;系统会认为用说明符/*……*/括起来的部分是说明部分,它可以是一行或多行内容,它们不参与程序的运行,仅起说明作用。注释文字可以是任意字符,如汉字、拼音、英文等。作为一个好的编程习惯,程序编制者在程序的起始部分应该写清楚该程序的一些基本特征,可以给编程者自己或其他程序阅读者起“望文知义”的作用。

另一个起注释作用的标志符是“//”,即两个斜杠,它表明本行中该符号后面的内容为注释内容,它同样也不会被编译到最终的可执行文件中。它与/*……*/的区别是前者仅作用于本行,而后者可以作用于多行。

(2) 注明使用库函数部分。它是以“#”标志的内容,用以标明在程序中使用到的函数库名称。每用到一个库函数就需要单独占有一行,并且以“#”开始。

从技术上讲,纯粹由程序员自己编写的语句构成C语言程序以完成某项功能是不可能的,但这却是罕见的。因为所有的C编译程序都提供能完成各种基本任务函数(如printf、scanf等)的集合——称之为函数库(如stdio.h等),程序员在编程过程中需要用到这些基本的函数时只需从函数库中直接调用即可。

当调用一个别人编写的函数时需要在程序开始时加以标明,表示该程序会使用到哪些函数库,函数库是用来标明的最小单位,哪怕只用到函数库中的一个函数,也需要将该函数库加以说明。在随后的编译与链接过程中,编译系统会自动寻找这些函数的代码并将它们与用户自己编写的代码结合起来,生成最终的可执行文件。

(3) 主函数main()。在C语言中,规定程序总是从main()函数开始执行,因此,该函数相当于程序的入口。main()函数中包含有用户自己编写的指令代码,也可以通过调用和控制其他函数进行协同工作,如上例中调用了printf()函数来实现在屏幕上打印字符的功能。需要强调的是,一个程序只能有一个main()函数。

一个C语言的函数由返回值、函数名、参数列表(或void表示没有返回值)和函数体组成,具体的结构将在后面章节加以详细描述。函数体的语法与main()函数的语法相同。

(4) 其他函数部分。如果在main()中需要调用其他的函数,而且这些函数不是系统提供的,就需要用户自己编写,这些函数可以接着main()函数书写,如Ex01_02.C所示。

```
/* 文件名: Ex01_02.C */
/* 功能: 输入长方体的长、宽、高, 计算长方体体积 */
#include<stdio.h>
void main( )
{
    int x,y,z,v ;                /*定义长、宽、高、体积为整型变量*/
    int volume(int a, int b, int c); /*定义被调用函数的类型*/
    printf("请输入长方体的三边长: "); /*显示输入提示*/
    scanf ("%d,%d,%d",&x,&y,&z); /*调用标准函数, 要求从键盘输入 x , y , z 的值*/
    v = volume(x,y,z);           /*调用 volume 函数, 计算体积*/
    printf("V = %d\n",v);        /*在屏幕上打印结果*/
}

int volume(int a, int b, int c) /*定义 volume 函数*/
{
    int p;                       /*定义函数内部使用的变量 p */
    p = a*b*c;                   /*计算体积 p 的值*/
    return(p);                  /*将 p 值返回调用处*/
}
```

运行时首先要求用户输入3个整数,然后程序自动计算出其体积,如图1-1所示。

由此可以看出:C程序为函数模块结构,所有的C程序都是由一个或多个函数构成,其中有且只有一个主函数main()。程序从主函数开始执行,当执行到调用函数的语句时,程序将控制转移到调用函数中执行,执行结束后,再返回主函数中继续运行,直至程序执行结束。



图1-1 程序Ex01_02运行结果

C程序的函数可以包括由编译系统提供的标准函数(如printf()、scanf()等)和由用户自己定义的函数。

1.3 C 语言的基本输入输出函数

1.3.1 格式化输入输出函数

C 语言标准函数库提供了两个标准的格式化输入、输出函数：printf()和 scanf()，这两个函数可以在标准输入输出设备上以各种不同的格式读写数据。

printf()函数用来向标准输出设备（显示屏幕）写数据，scanf()函数用来从标准输入设备（键盘）上读数据。下面将介绍这两个函数的具体用法。

1. printf()函数

printf()函数是格式化输出函数，一般用于向标准输出设备（如显示器）按规定格式输出信息。该函数的调用格式为：

```
printf("<格式化字符串>", <参量表>);
```

其中，<格式化字符串>包括两部分：一部分是正常字符，这些字符将按原样输出；另一部分是格式化规定字符，以“%”开始，后跟一个或几个规定字符，用来确定输出内容格式。

<参量表>是需要输出的一系列参数，其个数必须与格式化字符串所说明的输出参数个数相同，各参数之间用“,”分开，且顺序一一对应，否则将会出现意想不到的错误。

(1) 格式化规定符

C 语言提供的格式化规定符如表 1-1 所示。

表 1-1 格式化规定符

符 号	作 用
%d	十进制有符号整数
%u	十进制无符号整数
%f	浮点数
%s	字符串
%c	单个字符
%p	指针的值
%e	指数形式的浮点数
%x, %X	无符号以十六进制表示的整数
%o	无符号以八进制表示的整数
%g	自动选择合适的表示法

说明：

- 可以在“%”和字母之间插进数字表示最大场宽。

例如，%3d 表示输出 3 位整型数，不够 3 位右对齐；%9.2f 表示输出场宽为 9 的浮点数，其中小数位为 2 位，整数位为 6 位，小数点占 1 位，如果整个场宽不够 9 位则右对齐。%8s 表示输出 8 个字符的字符串，不够 8 个字符右对齐。

对于字符串或整型数据，如果字符串的长度或整型数位数超过说明的场宽，将按其实际长度

输出。

对于浮点数，若整数部分位数超过了说明的整数位宽度，将按实际整数位输出；若小数部分位数超过了说明的小数位宽度，则按说明的宽度四舍五入输出。

若想在输出值前加一些 0，就应在场宽项前加 0。例如，%04d 表示在输出一个长度小于 4 位的数值时将在前面补 0，使其总宽度为 4 位。

如果用浮点数表示字符或整型量的输出格式，小数点后的数字代表最大宽度，小数点前的数字代表最小宽度。例如，%6.9s 表示显示一个长度不小于 6 且不大于 9 的字符串，若大于 9，则第 9 个字符以后的内容将被删除。

- 可以在“%”和字母之间加小写字母 l，表示输出的是长型数。例如，%ld 表示输出长整数，%lf 表示输出双精度浮点数。
- 可以控制输出左对齐或右对齐。方法是在“%”和字母之间加入一个“-”号表示采用左对齐输出方式；如果不加则为右对齐输出方式。例如，%-7d 表示输出 7 位整数左对齐；%-10s 表示输出 10 个字符左对齐。

(2) 一些特殊规定字符

一些特殊规定字符如表 1-2 所示。

表 1-2 特殊规定字符	
符 号	作 用
\n	换行
\f	清屏并换页
\r	回车
\t	Tab 符
\xhh	表示一个 ASCII 码用十六进制表示，其中 hh 是 1 到 2 个十六进制数

下面是一个 printf()函数示例。

```
/*文件名: Ex01_03.C */
/*功能: 测试输出格式*/
#include<stdio.h>
#include<string.h>
#include<conio.h>
int main()
{
    char c, s[20], *p;
    int a=1234,d,*i;
    double f=3.141592653589;
    double x=0.12345678987654321;
    p="How do you do";
    strcpy(s, "Hello, World!");
    i = &d ;
    *i=12;
    c='\x41';
    printf("a=%d\n", a);           /*结果输出十进制整数      a=1234    */
    printf("a=%6d\n", a);         /*结果输出 6 位十进制数    a= 1234   */
    printf("a=%06d\n", a);        /*结果输出 6 位十进制数    a=001234  */
    printf("a=%2d\n", a);         /*a 超过 2 位，按实际值输出 a=1234    */
}
```

```

printf("*i=%4d\n", *i);      /*输出 4 位十进制整数      *i= 12      */
printf("*i=%-4d\n", *i);     /*输出左对齐 4 位十进制整数  *i=12      */
printf("i=%p\n", i);        /*输出地址                      */
printf("f=%f\n", f);         /*输出浮点数                    f=3.141593 */
printf("f=6.4f\n", f);       /*输出 6 位其中小数点后 4 位的浮点数 f=3.1416 */
printf("x=%lf\n", x);        /*输出长浮点数                  x=0.123457 */
printf("x=%18.16lf\n", x);    /*输出 18 位其中小数点后 16 位的长浮点数*/
printf("c=%c\n", c);         /*输出字符                      c=A      */
printf("c=%x\n", c);         /*输出字符的 ASCII 码值        c=41      */
printf("s[]=%s\n", s);       /*输出数组字符串              s[]=Hello, World!*/
printf("s[]=%6.9s\n", s);     /*输出最多 9 个字符的字符串    s[]=Hello, Wo */
printf("s=%p\n", s);         /*输出数组字符串首字符地址    s=FFBE    */
printf("*p=%s\n", p);        /* 输出指针字符串              p=How do you do */
printf("p=%p\n", p);        /*输出指针的值                  */
getch();
return 0;
}

```

运行结果如图 1-2 所示。

2. scanf()函数

scanf()函数是格式化输入函数，它表示从标准输入设备（键盘）读取输入的信息。该函数的调用格式为：

```
scanf("<格式化字符串>", <地址表>);
```

其中，<格式化字符串>包括以下 3 类不同的字符。

- 格式化说明符：格式化说明符与 printf()函数中的格式说明符基本相同。

- 空白字符：使 scanf()函数在读操作中略去输入中的一个或多个空白字符。

- 非空白字符：一个非空白字符会使 scanf()函数在读入时剔除掉与这个非空白字符相同的字符。

<地址表>是需要读入的所有变量的地址，而不是变量本身。这点与 printf()函数完全不同，需要特别引起注意。各个变量的地址之间用“,” 分开，如示例 1-1 所示。

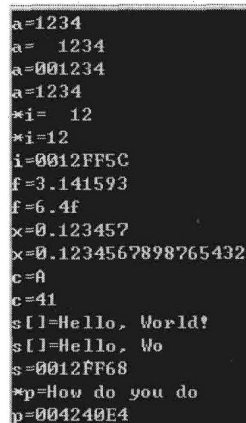
```

//示例 1-1
#include<stdio.h>
void main()
{   int i, j;
    printf("i, j=?\n");
    scanf("%d, %d", &i, &j);
}

```

该例中的 scanf()函数先读入一个整型数据，接着将输入的逗号剔除掉，最后读入另一个整型数据。如果“,” 这一特定字符没有找到，scanf()函数就终止运行。若格式符的参数之间的分隔符为一个空格，则从键盘输入参数的具体值时就在参数之间输入一个空格。有以下几点需要说明。

(1) 对于字符串数组或字符串指针变量，由于数组名和指针变量名本身就是地址，因此使用



```

a=1234
a= 1234
a=001234
a=1234
*i= 12
*i=12
i=0012FF5C
f=3.141593
f=6.4f
x=0.123457
x=0.1234567898765432
c=A
c=41
s[]=Hello, World!
s[]=Hello, Wo
s=0012FF68
*p=How do you do
p=004240E4

```

图 1-2 程序 Ex01_03 运行结果