



中等职业学校
21世纪计算机规划教材

C语言

程序设计

□ 刘怀亮 编著

冶金工业出版社



中等职业学校 21世纪计算机规划教材

- | |
|--|
| ☐ 计算机应用基础 |
| ☐ 汉字录入与编辑技术 |
| ☐ 常用工具软件 |
| ☐ 计算机专业英语 |
| ☐ 计算机网络基础 |
| ☐ 网页制作基础教程 |
| ☐ Internet 应用基础 |
| ☐ 数据库应用技术 (Access 2003 版) |
| ☒ C 语言程序设计 |
| ☐ C 语言程序设计上机指导与练习 |
| ☐ Visual Basic 语言程序设计 |
| ☐ 会计电算化基础 |
| ☐ 3ds max 7 三维动画基础与实例教程 |
| ☐ Photoshop CS 图像处理基础与实例教程 |
| ☐ 中文 AutoCAD 2005 实用教程 |
| ☐ CorelDRAW 12 平面设计基础与实例教程 |
| ☐ Flash MX 2004 动画制作基础与实例教程 |
| ☐ 计算机组装与维护教程 |
| ☐ 局域网组建与维护 |
| ☐ 数据库应用技术 (Visual FoxPro 6.0 版) |
| ☐ 多媒体应用技术 (Authorware 7.0 版) |
| ☐ 操作系统 |
| ☐ 微机原理与接口技术 |
| ☐ 办公自动化设备的使用与维护 |
| ☐ Word、Excel、PowerPoint 2003 实用教程 |
| ☐ 电子商务概论 |

ISBN 7-5024-3805-X



9 787502 438050 >

ISBN 7-5024-3805-X/TP·721

定价: 22.00 元

中等职业学校 21 世纪计算机规划教材

C 语言程序设计

刘怀亮 编著

北 京

冶金工业出版社

内 容 简 介

本书是作者在长期的教学和工程实践过程中，依据计算机专业相关教学大纲编写完成的，主要介绍了 C 语言的使用和程序设计的方法。全书共分为 13 章，主要内容包括：绪论，基本数据类型，运算符与表达式，顺序结构程序设计，选择结构程序设计，循环结构程序设计，函数，数组，指针，编译预处理，结构体、共同体和枚举类型，位运算，文件。

本书既可作为中等职业学校计算机技术专业的教材，也可作为计算机等级考试、计算机专业自学考试和各类计算机培训教材，同时也是计算机爱好者最好的 C 语言入门书。

图书在版编目 (CIP) 数据

C 语言程序设计 / 刘怀亮编著. —北京：冶金工业出版社，2005.8

ISBN 7-5024-3805-X

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

中国版本图书馆 CIP 数据核字 (2005) 第 080580 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号，邮编 100009)

责任编辑 程志宏

佛山市新粤中印刷有限公司印刷；冶金工业出版社发行；各地新华书店经销

2005 年 9 月第 1 版第 1 次印刷

787mm × 1092mm 1/16； 16.25 印张； 373 千字； 252 页

22.00 元

冶金工业出版社发行部 电话：(010) 64044283 传真：(010) 64027893

冶金书店 地址：北京东四西大街 46 号 (100711) 电话：(010) 65289081

(本社图书如有印装质量问题，本社发行部负责退换)

前 言

一、关于本书

在软件设计中，C 语言的使用占了很大一部分。C 语言是一种结构化、模块化、可编译的通用程序设计语言。它具有表达能力强、可移植性好和代码质量高等特点。C 语言兼有高级语言和低级语言的许多优点，已经成为目前国际上广泛应用的现代主流程序设计语言。掌握 C 语言的使用技巧，是软件开发人员最基本的技能。C 语言程序设计是计算机专业的一门重要课程。

本书是作者在长期的教学和工程实践过程中，依据计算机专业相关教学大纲编写完成的，主要介绍了 C 语言的使用和程序设计的方法。

二、本书结构

本书共分 13 章，具体结构安排如下：

第 1 章：绪论。主要介绍了 C 语言的发展史、特点、组成，Turbo C 基础以及 C 语言的术语词汇。

第 2 章：基本数据类型。主要介绍了标识符、常量和变量以及常用数据类型。

第 3 章：运算符与表达式。主要介绍了 C 语言运算符、表达式以及运算符的优先法和结合性。

第 4 章：顺序结构程序设计。主要介绍了 C 语言顺序结构程序设计以及相关实例分析。

第 5 章：选择结构程序设计。主要介绍了 C 语言选择结构程序设计以及相关实例分析。

第 6 章：循环结构程序设计。主要介绍了 C 语言循环结构程序设计以及相关实例分析。

第 7 章：函数。主要介绍了函数的调用、变量的作用范围与存储类别、内部函数和外部函数。

第 8 章：数组。主要介绍了 C 语言中的一维数组、字符数组、二维数组的定义及使用。

第 9 章：指针。主要介绍了 C 语言所特有的指针与指针变量、指针与函数、指针与数组等的应用。

第 10 章：编译预处理。主要介绍了宏定义、文件包含以及条件编译。

第 11 章：结构体、共同体和枚举类型。主要介绍了结构体、共同体和枚举类型的定义及使用。

第 12 章：位运算。主要介绍了 C 语言所特有的位运算、位运算符和位段的基础知识及使用。

第 13 章：文件。主要介绍了文件类型指针，文件的打开、关闭、读写、定位以及文件检测函数。

三、本书特点

本书是针对中等职业教育层次编写的，知识点较全面，深度适宜，语言简练，文笔流

畅，结构分明。为便于读者自学，每章后面都给出了练习题，使读者可以更深一步地理解 C 语言的应用，并能够在机器上练习和实践。

四、本书适用对象

本书的作者是计算机教学第一线的教师，在长期的教学实践过程中积累了丰富的教学实战经验，他们根据中等职业技术学校学生的认知规律和特点，总结归纳出一套理论联系实际的教学方法。在讲解各知识结构和技巧的同时，将重要的知识点融于实例中，这样，既便于教学又利于自学，为培养读者的灵活应用能力和创造能力奠定了基础。因此，本书既可作为中等职业学校计算机技术专业的教材，也可作为计算机等级考试、计算机专业自学考试和各类计算机培训教材，同时也是计算机爱好者最好的 C 语言入门书。

本书编写过程中得到许多同志的热情帮助，特别感谢刘启铭、蔡汉钧、陈晓聪和肖飞燕对本书所做的贡献，在此表示衷心感谢。

由于编写时间仓促、编者水平有限，书中不足之处在所难免，恳请广大读者朋友们批评指正。欢迎广大读者和专家对我们的工作提出宝贵建议，联系方式如下：

电子邮件：service@cnbook.net

网址：www.cnbook.net

本书附送的电子教案、代码及习题参考答案可从该网站的下载中心免费下载。此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者

2005 年 7 月

目 录

第 1 章 绪论	1	3.1.1 运算符的种类	31
1.1 概述	1	3.1.2 表达式的概念	32
1.1.1 C 语言发展简史	1	3.2 算术运算符与算术表达式	32
1.1.2 C 语言的特点	2	3.2.1 五种基本的算术运算符	32
1.1.3 C 程序的组成	3	3.2.2 算术表达式	34
1.2 Turbo C 简介	5	3.3 赋值运算符与赋值表达式	34
1.2.1 Turbo C 2.0 的安装和启动	6	3.3.1 赋值运算符	34
1.2.2 运行一个 C 语言程序的 一般过程	7	3.3.2 类型转换	34
1.2.3 Turbo C 2.0 主菜单的介绍	8	3.3.3 复合赋值运算符	35
1.3 C 语言的术语词汇	13	3.3.4 赋值表达式	36
小结	14	3.4 关系运算符与关系表达式	36
习题一	14	3.4.1 关系运算符	36
一、选择题	14	3.4.2 比较运算符	37
二、填空题	15	3.4.3 不等于运算符	37
三、综合题	15	3.4.4 嵌入赋值	37
第 2 章 基本数据类型	16	3.4.5 关系表达式	38
2.1 概述	16	3.5 逻辑运算符与逻辑表达式	38
2.2 标识符、常量和变量	17	3.5.1 逻辑运算符及其运算规则	38
2.2.1 标识符	18	3.5.2 逻辑非运算符	39
2.2.2 常量	18	3.5.3 逻辑与和逻辑或运算符	39
2.2.3 变量	19	3.5.4 逻辑运算的运算优先级	39
2.3 整型数据	20	3.5.5 逻辑表达式	40
2.4 浮点型数据	22	3.6 其他运算符	40
2.5 字符型数据	23	3.6.1 自增、自减运算符	40
2.6 各种基本数据类型的混合运算	27	3.6.2 逗号运算符与逗号表达式	41
小结	29	3.6.3 条件运算符与条件表达式	42
习题二	29	3.7 运算符的优先级和结合性	43
一、选择题	29	小结	44
二、填空题	30	习题三	44
三、综合题	30	一、选择题	44
第 3 章 运算符与表达式	31	二、填空题	45
3.1 概述	31	三、综合题	46
3.1.1 运算符的种类	31	第 4 章 顺序结构程序设计	48
3.1.2 表达式的概念	32	4.1 C 语句概述	48
3.2 算术运算符与算术表达式	32		
3.2.1 五种基本的算术运算符	32		
3.2.2 算术表达式	34		
3.3 赋值运算符与赋值表达式	34		
3.3.1 赋值运算符	34		
3.3.2 类型转换	34		
3.3.3 复合赋值运算符	35		
3.3.4 赋值表达式	36		
3.4 关系运算符与关系表达式	36		
3.4.1 关系运算符	36		
3.4.2 比较运算符	37		
3.4.3 不等于运算符	37		
3.4.4 嵌入赋值	37		
3.4.5 关系表达式	38		
3.5 逻辑运算符与逻辑表达式	38		
3.5.1 逻辑运算符及其运算规则	38		
3.5.2 逻辑非运算符	39		
3.5.3 逻辑与和逻辑或运算符	39		
3.5.4 逻辑运算的运算优先级	39		
3.5.5 逻辑表达式	40		
3.6 其他运算符	40		
3.6.1 自增、自减运算符	40		
3.6.2 逗号运算符与逗号表达式	41		
3.6.3 条件运算符与条件表达式	42		
3.7 运算符的优先级和结合性	43		
小结	44		
习题三	44		
一、选择题	44		
二、填空题	45		
三、综合题	46		

4.2 数据的输入与输出	50	6.6.1 break 语句	84
4.2.1 数据输入/输出的概念及在 C 语言中的实现	50	6.6.2 continue 语句	84
4.2.2 格式化输入/输出函数	50	6.6.3 goto 语句	85
4.2.3 字符输入/输出函数	59	6.7 循环结构程序设计举例	85
4.3 顺序结构程序设计举例	60	小结	87
小结	61	习题六	88
习题四	62	一、选择题	88
一、选择题	62	二、填空题	88
二、填空题	63	三、综合题	89
三、综合题	63	第 7 章 函数	91
第 5 章 选择结构程序设计	65	7.1 概述	91
5.1 概述	65	7.1.1 函数的定义	93
5.2 if 语句	65	7.1.2 函数的参数	94
5.2.1 if 语句的一般应用	66	7.1.3 函数的返回值	95
5.2.2 if 语句的嵌套与嵌套匹配原则	67	7.2 函数的调用	96
5.2.3 条件运算符	69	7.2.1 函数的声明	96
5.3 switch 语句	70	7.2.2 函数调用的数据传递方式	98
5.3.1 switch 结构的一般应用	70	7.2.3 函数的嵌套调用和递归调用	100
5.3.2 switch 语句的嵌套	71	7.3 变量的作用范围	106
5.3.3 if 语句和 switch 语句的 相互转换	72	7.3.1 局部变量	106
5.4 选择结构程序设计举例	72	7.3.2 全局变量	107
小结	74	7.4 变量的存储类别(选学)	109
习题五	74	7.4.1 内部变量	109
一、选择题	74	7.4.2 外部变量	109
二、填空题	75	7.4.3 其他变量	109
三、综合题	76	7.5 内部函数和外部函数	115
第 6 章 循环结构程序设计	78	7.5.1 内部函数	115
6.1 概述	78	7.5.2 外部函数	115
6.2 while 和 do-while 语句	78	7.6 典型实例剖析	116
6.2.1 while 语句	78	小结	118
6.2.2 do-while 语句	79	习题七	119
6.3 for 语句	81	一、选择题	119
6.4 循环结构的嵌套	83	二、填空题	120
6.5 几种循环语句的比较	83	三、综合题	121
6.6 其他语句	84	第 8 章 数组	123
		8.1 概述	123
		8.2 一维数组	123

8.2.1 一维数组的定义	123	二、填空题	160
8.2.2 一维数组的引用	124	三、综合题	160
8.2.3 一维数组的赋值	125		
8.2.4 应用举例	126		
8.3 字符数组	127	第 10 章 编译预处理	162
8.3.1 字符数组和字符串数组的 定义	127	10.1 概述	162
8.3.2 字符数组的基本操作和整体 操作	128	10.2 宏定义	162
8.3.3 常用字符串处理函数	128	10.2.1 不带参数的宏定义	162
8.3.4 应用举例	131	10.2.2 带参数的宏定义	165
8.4 二维数组	132	10.3 文件包含	168
8.4.1 二维数组概述	132	10.4 条件编译	170
8.4.2 二维数组中各元素的引用	132	10.4.1 第一种条件编译形式	170
8.4.3 二维数组元素的地址	133	10.4.2 第二种条件编译形式	171
8.4.4 二维数组的初始化	134	10.4.3 第三种条件编译形式	171
8.4.5 应用举例	134	小结	172
8.5 数组和函数	135	习题十	172
8.6 典型例题剖析	137	一、选择题	172
小结	139	二、填空题	173
习题八	139	三、综合题	174
一、选择题	139		
二、填空题	140	第 11 章 结构体、共同体和枚举类型	175
三、综合题	140	11.1 概述	175
第 9 章 指针	143	11.1.1 结构体的定义	175
9.1 概述	143	11.1.2 结构体变量的定义	177
9.2 指针与指针变量	143	11.1.3 结构体成员的引用与结构体 变量的初始化	178
9.2.1 地址的概念	143	11.1.4 结构体数组	180
9.2.2 指针的定义	144	11.1.5 指向结构体类型数据的指针	183
9.2.3 指针变量的定义	144	11.2 链表	187
9.2.4 指针变量的使用	144	11.2.1 概述	187
9.3 指针与函数	148	11.2.2 链表的基本概念	188
9.4 指针与数组	150	11.2.3 建立链表	189
9.5 指针的其他用法	156	11.2.4 向链表中插入、删除结点	191
9.6 典型例题剖析	156	11.3 共同体	196
小结	158	11.3.1 共同体类型和共同体 变量的定义	197
习题九	159	11.3.2 共同体变量引用	198
一、选择题	159	11.4 枚举类型	200
		11.4.1 枚举类型和枚举变量的定义	200
		11.4.2 枚举变量的使用	200

小结	202	13.5 文件的读写	220
习题十一	202	13.5.1 读字符函数 fgetc	221
一、选择题	202	13.5.2 写字符函数 fputc	222
二、填空题	204	13.5.3 读字符串函数 fgets	223
三、综合题	204	13.5.4 写字符串函数 fputs	224
第 12 章 位运算	206	13.5.5 数据块读写函数 fread 和 fwrite	225
12.1 概述	206	13.5.6 格式化读写函数 fscanff 和 fprintf	226
12.2 位运算和位运算符	206	13.6 文件的定位	227
12.2.1 按位与	207	13.7 文件检测函数	228
12.2.2 按位或	208	小结	228
12.2.3 按位异或	209	习题十三	230
12.2.4 求反	210	一、选择题	230
12.2.5 左移	211	二、填空题	231
12.2.6 右移	212	三、综合题	231
12.2.7 位运算赋值运算符	212	附录	233
12.3 位段	213	A.1 常用字符与 ASCII 代码对照表	233
小结	214	A.2 C 语言中的保留字	233
习题十二	215	A.3 C 语言的基本数据类型的 分类及特点	234
一、选择题	215	A.4 常用的转义字符及其含义	234
二、填空题	215	A.5 运算符的优先级和结合性	234
三、综合题	216	A.6 C 语言的标准库函数	236
第 13 章 文件	217	A.7 Turbo C (V2.0) 编译错误信息	246
13.1 概述	217	参考文献	252
13.2 文件类型指针	218		
13.3 文件的打开	218		
13.4 文件的关闭	220		

第1章 绪 论

学习目标:

- (1) 了解 C 语言的发展历史。
- (2) 了解 C 语言的特点。
- (3) 理解组成 C 语言的各个部分。
- (4) 学会使用 Turbo C 2.0 编辑 C 程序。

1.1 概述

1.1.1 C 语言发展简史

C 语言是国际上最著名的高级程序设计语言之一,是当今世界上使用范围最广的一种计算机编程语言。它不仅适合作为系统描述语言,用来写系统软件,也可以用来写应用软件。在 C 语言诞生以前,系统软件主要是用汇编语言编写的。由于汇编语言高度依赖于计算机硬件,用它编写的程序的可读性和可移植性都比较差。要提高程序的可读性和可移植性,最好改用高级语言编写代码,但一般高级语言难以实现计算机硬件的直接操作(这正是汇编语言的优势),于是人们盼望有一种兼有低级语言特性的高级语言,C 语言就在这种情况下应运而生了。

C 语言是在 B 语言的基础上发展起来的,它的根源可以追溯到 ALGOL 60。1960 年推出的 ALGOL 60 是一种面向问题的高级语言,它离硬件比较远,不宜用来编写系统程序。1963 年英国的剑桥大学推出了 CPL (Combined Programming Language) 语言。CPL 语言在 ALGOL 60 的基础上更接近硬件一些,但规模比较大,难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 语言做了简化,推出了 BCPL (Basic Combined Programming Language) 语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,又做进一步简化,设计出了很简单的而且很接近硬件的 B 语言(取 BCPL 的第一个字母),并用 B 语言写了第一个 UNIX 操作系统,在 PDP-7 上实现。1971 年在 PDP-11/20 上实现了 B 语言,并重写了 UNIX 操作系统。但 B 语言过于简单,功能有限。1972 年至 1973 年间,贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母)。C 语言既保持了 BCPL 和 B 语言的优点——精练,接近硬件,又克服了它们的缺点——过于简单,数据无类型等。最初的 C 语言只是为描述和实现 UNIX 操作系统而设计的一种工作语言。1973 年, K.Thompson 和 D.M.Ritchie 两人合作把 UNIX 的 90%以上用 C 改写,即 UNIX 第 5 版。原来的 UNIX 操作系统是 1969 年由美国贝尔实验室的 K.Thompson 和 D.M.Ritchie 开发成功的,是用汇编语言写的。

C 语言的演变过程如下所示:

ALGOL 60(1960 年)→CPL(1963 年)→BCPL(1967 年)→B(1970 年)→C(1972 年)

在初期阶段, C 语言还仅限于在贝尔实验室内部使用。到 1975 年,随着 UNIX 第 6 版问世并免费推向世界, C 语言的一系列优点才引起人们的广泛关注。随着 UNIX 系统从

研究室、院校走向商业社会，登上世界舞台，C 语言也逐渐为世人所认识、欢迎和使用；特别是“可移植 C 语言编译程序”在 1977 年推出之后，推动了 UNIX 操作系统在各种机器上的实现。时至今日，C 语言已独立于 UNIX 系统，独立于 PDP-11 而发展起来了。C 语言是近二十年来对计算机程序设计作出重大贡献的一种程序设计语言。它已成为微型、小型、中型、大型和超大型（巨型）计算机上共同使用的一种程序设计语言。

1983 年 K.Thompson 和 D.M.Ritchie 两人同时获得了 ACM 图灵奖，这是计算机科学和技术领域的最高荣誉奖，以表彰他们在开发 UNIX 系统和 C 语言方面所做出的杰出贡献。

1978 年 B.W.Kernighan 和 D.M.Ritchie 合著了《The C Programming Language》一书，该书中讲述的 C 语言为后来被广泛使用的 C 语言的范本，它被称为标准 C。根据 C 语言问世以来的各种版本对 C 的发展和扩充，1983 年美国国家标准化协会（ANSI）制定了新的标准，称为 ANSI C。ANSI C 与标准 C 相比，有了很大的发展。B.W.Kernighan 和 D.M.Ritchie 在 1988 年对他们的经典著作《The C Programming Language》进行了修改，使之符合 ANSI C 标准。1990 年，国际标准化组织 ISO（International Standard Organization）通过了 C 程序设计语言的国际标准（ISO 9899 - 1990），它是以 ANSI C 为基础，吸收了国际上的意见后制定的。从此，C 语言就有了统一的国际标准，这标志着 C 语言进入一个崭新的时代。

1.1.2 C 语言的特点

C 语言能成为深受广大程序设计人员欢迎的主力编程语言，并不是取决于其商品推销策略的正确，而是由于它本身具有很显著的优点所展示出的强大的生命力。C 语言的主要特点有：

（1）语言表达能力强。C 语言是面向结构程序设计的语言，有良好的通用性，可以在各种硬件平台上运行。它可以直接处理字符、数字和地址，可以完成通常由硬件设备实现的算术、逻辑运算，可以充分地反映出当前计算机的性能，能取代汇编语言编写各种系统软件和应用软件。由于 C 语言既具有高级语言的功能，又具有低级语言的很多特性，所以人们往往把 C 语言称为“中级语言”。它表明 C 语言能把高级语言的结构化与低级语言的实用性两者有机地结合起来。

（2）语言简洁，使用方便、灵活。C 语言在表达方式上力求简单易行，仅使用一些简单规则和方法构成相当复杂的数据类型、语句和程序结构。另外，C 语言把一般语言的许多功能模块都通过显式函数调用来完成，使得编译程序相对小而精。例如，C 语言本身没有提供输入/输出机制，也没有并行操作，同步或协同程序等复杂控制，而是提供了大量而有效的库函数来实现输入/输出、字符处理及存储分配等功能。库函数可根据需要方便地扩充，这样就使得 C 语言编译程序小巧、紧凑。

（3）运算符丰富。C 语言是一种表达式语言，它有功能很强的运算符。C 语言的运算符（见附录）包含的范围很广泛，多达 44 个。C 把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 运算类型极其丰富，表达式类型灵活、多样，在其他高级语言中难以实现的运算在 C 语言中都很容易办到。

（4）数据结构丰富，具有现代化语言的各种数据结构。C 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。能用来实现各种复杂的数据结构（如链表、树、栈等）的运算，尤其是指针类型数据，使用起来灵活、多样。

(5) C 语言是一种结构化程序设计语言, 特别适合于大型程序的模块化设计; C 语言具有编写结构化程序所必需的基本流程控制语句。C 语言程序是由集合构成的, 函数各自独立作为模块化设计的基本意念。它所包含的源文件, 可以分割成多个源程序, 分别对其进行编译, 然后连接起来构成可执行的目标文件。C 语言还提供了多种存储属性, 可以使数据按其需要在相应的作用域内起作用。

(6) C 语言为字符、字符串、集合和表的处理提供了良好的基础, 它能够表示和识别各种可显示的及能起控制作用的字符, 也能够区分和处理单个字符和字符串。

(7) 语法限制不大, 程序设计自由度大。一般的高级语言编译时语法检查比较严, 能检查出几乎所有的语法错误。而 C 语言程序编写者有较大的自由度, 因此放宽了语法检查。程序员应当仔细检查程序, 保证其正确, 而不要过分依赖 C 编译程序去查错。

(8) 生成的目标代码质量高, 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%。

(9) 具有预处理程序和预处理语句, 给大型程序的编写和调度提供了方便。

(10) 具有良好的移植性(与汇编语言相比)。用 C 语言编写的模块基本上不做修改就能用于各种型号的计算机和各种操作系统。

上面介绍了 C 语言的一般特点, 它的其他内部特点将在具体章节中予以说明。读者也可在今后的学习和实践中深入体会, 自行进行归纳、总结。总之, C 语言是程序设计人员的语言, 它使用灵活, 限制少, 功能模块化, 接近汇编语言的效率, 可在不同平台间移植, 这一切都使得 C 语言成为最受专业程序设计人员所喜爱的语言, 使得 C 语言的应用面越来越广。现在, C 语言已不仅用来编写系统软件, 也用来编写应用软件。学习和使用 C 语言的人已越来越多。

1.1.3 C 程序的组成

为了使读者对 C 语言程序有一个感性认识, 下面用几个简单的 C 程序作示例, 从中分析 C 程序的特性。

【例 1-1】 程序运行后显示一句话: Welcome you!

```
main( )
{
    printf("Welcome you!\n");
}
```

这个程序运行以后, 就输出以下一行信息:

```
Welcome you!
```

其中 main() 是函数首部, 它告诉系统, 这是一个函数, 名字为 main。main 函数是 C 语言中标识主函数的专用名, 每一个 C 程序都必须有一个 main 函数, 表示该 C 程序从这里开始执行。函数体由大括号 { } 括起来。本例中主函数内只有一个输出语句, printf 是 C 语言中标准输出函数, 这里对它进行调用。双引号(双撇号)内的字符串照原样输出。“\n”是由反斜线“\”和字母“n”构成, 合起来作为一个换行符。当遇到换行符时, 光标或打印机就移到下行的开头。printf() 语句的分号是语句结束符, 这是 C 语言的约定: 分号是语句结束符, 是语句的一个组成部分。在 C 语言程序中, 一个语句结束必须有一个分号。

【例 1-2】 计算两个整数的和。

```
main( )
{
```

```

int a,b,sum;           /*定义变量*/
a=135;b=246;          /*以下3行为C语句*/
sum=a+b;
printf("sum is %d\n",sum);
}

```

本程序的作用是求两个整数 a 和 b 之和 sum。a 的值是 135，b 的值是 246，程序运行后，输出结果是：

```
sum=381
```

/*.....*/表示注释部分，在这里用汉字表示注释是为了方便理解，当然也可以用英语或汉字拼音作注释。注释的目的是为了增加程序的可读性。它是编程人员对整个程序或者某些特定的程序段以及某些重要的数据加以说明，对编译和运行不起作用。注释可以加在程序中任何位置。第 3 行是声明部分，定义 a, b, sum 三个整型变量。第 4 行是两个赋值语句使 a 和 b 的值分别为 135 和 246。第 5 行使 sum 的值为 a+b。第 6 行是函数调用语句，调用标准输出函数 printf，按指定格式输出运算结果。其中，“%d”是输入/输出的“格式字符串”，用来指定输入/输出时的数据类型和格式（详见第 4 章），“%d”表示“十进制整数类型”。在执行该函数调用时，将后面对应参数 sum 的值按十进制整数形式输出；而“sum=”普通字符串，按原样印出。

【例 1-3】比较两个数的大小，并输出最大的数。

```

main( )                /*主函数*/
{
    int a,b,c;           /*声明部分，定义变量*/
    scanf ("%d,%d",&a,&b); /*输入变量 a 和 b 的值*/
    c=max (a,b);         /*调用 max 函数，将得到的值赋给 c*/
    printf ("max=%d",c); /*输出 c 的值*/
}
int max(int x,int y)     /*定义 max 函数，函数值为整型，形式参数 x, y 为整型*/
{
    int z;              /*max 函数中的声明部分，定义本函数中用到的变量 z 为整型*/
    if (x>y) z=x;
    else z=y;
    return (z);         /*将 z 的值返回，通过 max 带回调用处*/
}

```

本程序包括两个函数：主函数 main 和被调用的函数 max。max 函数的作用是将 x 和 y 中较大者的值赋给变量 z。将 return 语句的值返回给主调函数 main。返回值是通过函数名 max 带回到 main 函数的调用处。Main 函数中的 scanf 是“输入函数”的名字（scanf 和 printf 都是 C 语言提供的标准输入/输出函数）。程序中 scanf 函数的作用是输入 a 和 b 的值。&a 和 &b 中的“&”的含义是“取地址”，此 scanf 函数的作用是将两个数值分别输入到变量 a 和 b 的地址所标志的单元中，也就是输入给变量 a 和 b。关于 scanf 函数详见第 4 章。

main 函数中第 5 行为调用 max 函数，在调用时将实际参数 a 和 b 的值分别传送给 max 函数中的形式参数 x 和 y。经过执行 max 函数得到一个返回值（即 max 函数中变量 z 的值），把这个值赋给变量 c，然后输出 c 的值。printf 函数中双引号内的“max=%d”，在输出时，其中“%d”将由 c 的值取代之，“max=”原样输出。

程序运行情况如下：

```

10, 3 /                (输入 10 和 3 给 a 和 b)
max=10                 (输出 c 的值)

```

本例用到了函数调用，实际参数和形式参数等概念，这里只做了很简单的解释。读者如对此不大理解，可以先不予深究，在学到以后有关章节时，问题自然迎刃而解了。在此介绍的例子，无非是使读者对 C 程序的组成和形式有一个初步了解。

通过这几个例子可以看到：

(1) 函数是 C 语言程序的基本单位。一个 C 语言程序是由一个或多个函数所组成。C 语言程序中必须有一个名为 `main` 的主函数，其余函数的名字由程序设计者自定。`main()` 函数的作用相当于其他高级语言中的主程序；其他函数的作用相当于子程序。

(2) C 语言程序总是从 `main()` 函数开始执行。一个 C 语言程序，总是从 `main()` 函数开始执行，而不论其在程序中的任何位置。当主函数执行完毕时，亦即程序执行完毕。

习惯上，将主函数 `main()` 放在最前面。

(3) 任何函数（包括主函数 `main()`）都是由函数说明和函数体两部分组成。其一般结构如下：

[函数类型]	函数名(函数参数表)	函数说明部分
	{ 局部变量说明语句部分; 执行语句部分;	函数体部分
	}	

函数说明指明函数的类型、属性、函数名、函数参数和参数说明等。

一个函数名后面必须跟一对圆括号，函数参数可以没有，如 `main()`。

函数体是大括号所括的部分，它包括局部变量的说明语句和一组执行语句。当然在某种情况下也可以没有声明部分。甚至可以既无声明部分，也无执行部分。如：

```
dump( )  
{ }
```

它是一个空函数，什么也不干，但这是合法的。

(4) C 程序书写格式自由，一行内可以写几个语句，一个语句可以书写在多行上。C 程序没有行号，也不像 FORTRAN 或 COBOL 那样严格规定书写格式（语句必须从某一列开始书写）。

(5) 每个语句和数据定义的最后必须有一个分号。分号是 C 语句不可缺少的组成部分。如：

```
c=a+b;
```

即使是程序中最后一个语句也应包含分号。

(6) C 语言本身没有输入/输出语句，输入和输出的操作是由库函数 `scanf` 和 `printf` 等函数来完成的，C 对输入/输出操作实行“函数化”。由于输入/输出操作牵涉到具体的计算机，把输入/输出操作放在函数中处理，就可以使 C 语言本身的规模较小，编译程序简单，很容易在各种机器上实现，程序具有可移植性。

(7) C 程序中的注释是以 `/*` 开头、`*/` 结束的任意字符串，用于对程序功能、算法、数据等进行说明，提高程序的可读性。注释在程序中的位置是任意的。但 C 程序的注释不允许嵌套出现。

1.2 Turbo C 简介

所谓程序设计，就是根据计算机要完成的任务，提出相应的要求，在此基础上设计数据结构和算法，然后再编写相应的程序代码并测试该代码运行的正确性，直到能够得到正确的运行结果为止。

在计算机中，各种信息和数据都是以文件形式存放的。在编辑方式下建立起来的程序文件称为源程序文件，简称源文件（如 `f.c`），相应的程序叫做源程序。源程序是用高级语言编写的，它不能直接在机器上运行。因为计算机只能识别和执行由 0 和 1 组成的二进制的指令。为了使计算机能执行高级语言源程序，必须先用一种称为“编译程序”的软件，把

源程序翻译成二进制形式的“目标程序”，然后将该目标程序与系统的库函数和其他目标程序连接起来，形成可执行的目标程序。

对于一个具体给定的问题，首先是写出 C 源程序，然后是上机调试、运行，直到得到满意的结果，这中间要经过若干步骤。一般说来，这些步骤有：上机输入与编辑源程序→对源程序进行编译→与库函数连接→运行目标程序。以上过程如图 1-1 所示。

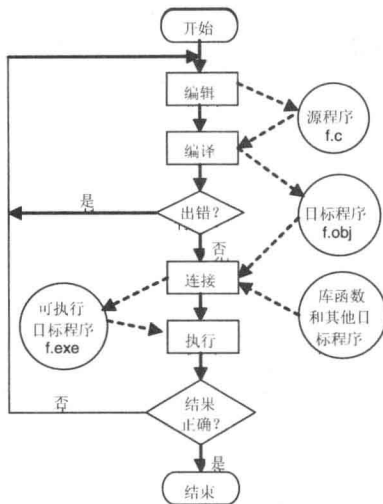


图 1-1 程序设计步骤示意图

其中实线表示操作流程，虚线表示文件的输入/输出。例如，编辑后得到一个源程序文件 f.c，然后再将源程序文件 f.c 输入编译器，经过编译得到目标程序文件 f.obj，再将目标程序文件 f.obj 输入内存，与系统提供的库函数等连接，得到可执行的目标程序文件 f.exe，最后把 f.exe 调入内存并运行。为方便读者上机练习，下面给出在 Turbo C 环境下运行 C 程序的简要步骤。

Turbo C 是美国 Borland 公司的产品，Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件，如 Turbo BASIC，Turbo Pascal，Turbo Prolog，这些软件很受用户欢迎。Turbo C 是一个集成开发环境，把程序的编辑、编译、连接、运行等操作集中在一个界面上进行，使用十分方便。下面就着重介绍 Turbo C 2.0 的用法。

1.2.1 Turbo C 2.0 的安装和启动

目前常用的 Turbo C 2.0 一般情况下是不需要安装的。Turbo C 2.0 系统经过压缩之后只有 1MB 左右，如果是压缩文件，先在硬盘上创建一个新目录，可将压缩文件直接解压到 c:\tc20 新目录下或其他路径，如果不是压缩文件，可直接复制到 c:\tc20 目录下即可。运行 Turbo C 2.0 时，常用的方法有以下几种：

(1) 在 DOS 状态下，可以切换到 c:\tc20，然后键入 tc 按回车键即可，不过这种方法是不可取的，因为这样产生的 .c, .exe, .obj 文件会存放在此目录下，给管理带来很大麻烦。

(2) 在 Windows 环境下，用户可以在桌面上建立一个快捷方式，指向 c:\tc20\tc.exe，这时运行 Turbo C 2.0 比较方便，双击快捷图标即可。