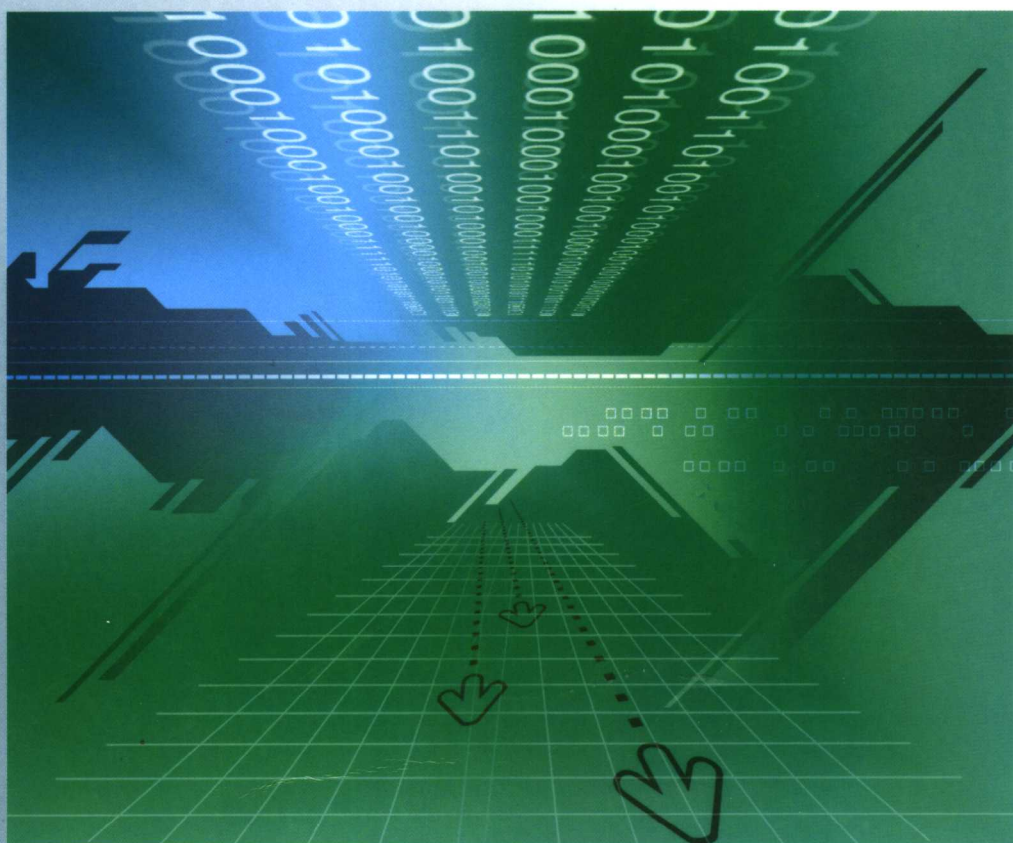




21 世纪高职高专计算机通用规划教材

C 语言程序设计

朱云洁 邓新治 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

 21 世纪高职高专计算机通用规划教材

C 语言程序设计

主 编 朱云洁 邓新治

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是 21 世纪职业教育教学改革计算机基础课程的标准教材,教材以讲授与自学结合的风格编写,适合于各类职业院校的学生使用。内容包括 C 语言的语法规则、各种数据类型、语句格式和函数结构等。

本书始终贯彻分层教学思想,针对不同水平的学生,设计了不同的学习方式。一开始就从 C 语言程序的基本特点入手,先认识 C 语言程序,逐步通过具体问题的需要介绍 C 语言的命令、语句和数据类型,通过“实践—理论—再实践”,使学生具有带着问题学习,以提高学习的兴趣和学习效率。

全书系统性强,循序渐进、重点突出,采用边讲边练、边用边学的方式,每个知识点都配有思考题,便于初学者掌握,又便于专业人员提高,更便于教师授课。教材还有专门配套的上机实训教材,方便学生课外上机操作时使用。

图书在版编目(CIP)数据

C 语言程序设计/朱云洁,邓新治主编. —北京:中国铁道出版社,2006.8
21 世纪高职高专计算机通用规划教材
ISBN 7-113-07194-5

I. C... II. ①朱...②邓... III. C 语言—程序设计
—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 103077 号

书 名: C 语言程序设计

作 者: 朱云洁 邓新治 等

出版发行: 中国铁道出版社(100054,北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 孙文娟

责任编辑: 苏 茜 翟玉峰

特邀编辑: 薛秋沛 李成都

封面设计: 薛 为

封面制作: 白 雪

责任校对: 陈 宏

印 刷: 北京鑫正大印刷有限公司

开 本: 787×1092 1/16 印张: 13 字数: 301 千

版 本: 2006 年 8 月第 1 版 2006 年 8 月第 1 次印刷

印 数: 1~4 000 册

书 号: ISBN 7-113-07194-5/TP·1908

定 价: 22.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

前 言

信息时代的到来将是人类社会的又一场意义深远的革命，它必将对社会、政治、经济、军事、文化和人们的思想观念、生活、工作、学习及思维方式等带来深刻的变革，也必将对教育事业产生巨大的影响。如何更好地把握时代脉搏、探索育人之路，是摆在学校教学工作面前的重大课题。

今天，这场革命终于发生了，这是一场在全世界范围内悄然发生的全面的学习革命。这场革命的目标并不是改变现行的教育体制和学校的教育方式，它的目标要深远得多，它还要改变我们的家庭教育方式和员工培训方式，改变我们整个的学习方式、生活方式和工作方式，并最终改变我们在未来的生存方式。我们先从这里找到突破口，以赶上信息时代的步伐。

教材是知识传授和能力培养的基础，C 语言比较复杂，规则较多、使用灵活、容易出错，不少初学者都感到困难。本书力求让读者花最少的时间学会最多的东西，这是编写本教材的根本目的。

教材从学生的实际出发，以能力为本位，深入浅出，打破传统教材的编排结构，将理论与实践有机地结合在一起，融知识传授和能力培养于一体，仅用一个课时就能完整地编写一个程序，始终体现了一个“新”字，所以它又是一个学习 C/C++ 语言的“直通车”，不管读者的文化程度如何，它都能把读者带到目的地。

本书的特色之一是例题丰富，思考题典型，检测点灵活，实训内容充实、涉及面广。我们力求通过实际例题的讲解，逐步提高学生的编程思想；通过思考题提升学生的知识水平；通过检测点巩固和消化知识点的内涵；通过上机实训强化学生的编程能力。

本书还专门设计了一些应用性的章节，如图形模式和文本模式及动画技术等，学生在学习的过程中可根据自己的兴趣爱好进行选择，并在每个知识点中都设有不同的学习内容，可供不同程度和不同精力的学生选学（注：章节前加“△”的为自学内容，加“*”的为选学内容，教师在授课的过程中可根据情况进行讲授）。

编者作为教师和教研工作者，且作为较早进行计算机程序设计应用的研究者，了解学生、计算机应用人员的需求，为此，将自己的一些实践、体会、心得及一些新思想写到本书中，奉献给读者，以期对教学工作者及编程人员有所帮助。

本书由朱云洁、邓新治主编，高卫卫、宗锐、刘斌、高阿云、聂璐等参与编写，并有配套的《C 语言程序设计实验指导》。

感谢广大朋友对完成本书所给予的帮助，书中难免有错误和不妥之处，恳请同行和广大读者批评指正。

编 者

2006 年 5 月 5 日

目 录

第 1 章 绪 论	1
1.1 C 语言概述	2
1.2 C 和 C++	2
1.3 为什么学习 C	3
1.4 几个基本概念	3
△1.5 C++ 的特点	4
第 2 章 Turbo C 开发环境	5
2.1 Turbo C 概述	5
2.2 Turbo C 开发环境介绍	8
2.2.1 启动 TurboC	8
2.2.2 几个典型操作	9
2.2.3 退出 Turbo C	10
2.2.4 装入 C 源文件	10
2.2.5 编译、运行和调试	10
2.2.6 编辑窗口和信息窗口	10
2.2.7 File 菜单中的各项	11
2.2.8 剪辑窗口——Clipboard	11
第 3 章 C 语言概述	14
3.1 第一个 C 程序	14
3.2 标识符、常量、变量、运算符和表达式	15
3.2.1 标识符	15
3.2.2 数据类型和常量、变量	15
3.2.3 运算符	16
3.2.4 表达式	17
3.3 函数的基本概念	18
3.3.1 C 语言中的函数	18
3.3.2 函数和变量	19
3.3.3 全局变量和局部变量	19
3.3.4 返回值	20
3.4 分支控制语句	20
3.4.1 if 语句	23
3.4.2 开关语句 switch	27
3.5 循环控制语句	30
3.5.1 循环示例	30
3.5.2 for 循环	33

3.5.3 while 循环.....	34
3.5.4 do...while 循环.....	36
3.6 综合应用举例.....	38
第 4 章 数组和字符串.....	45
4.1 一维数组.....	45
4.1.1 一维数组的基本应用.....	45
4.1.2 数组的操作和初始化.....	46
4.1.3 数组的操作.....	47
4.2 字符串.....	49
4.2.1 字符串作为数组.....	49
4.2.2 字符串的初始化.....	49
4.2.3 字符串的输入/输出.....	50
4.2.4 用于字符串的几个函数.....	51
4.3 二维数组.....	54
4.3.1 二维数组示例.....	54
4.3.2 字符串数组.....	59
4.4 多维数组.....	60
4.4.1 字符数组.....	60
4.4.2 字符串（数组）的处理.....	61
△第 5 章 文本模式.....	67
5.1 文本模式概述.....	67
5.1.1 文本模式的概念.....	67
5.1.2 文本窗口.....	68
5.2 默认窗口的操作.....	68
5.3 屏幕的文本移动与存取函数.....	69
5.3.1 屏幕文本移动函数.....	69
5.3.2 屏幕文本存取函数.....	69
第 6 章 程序调试与测试.....	71
6.1 程序的正确性.....	71
6.1.1 程序的正确性概述.....	71
6.1.2 程序的错误.....	71
6.2 程序的调试.....	72
6.2.1 调试的概念.....	72
6.2.2 常见的调试方法.....	72
6.3 Turbo C/Borland C++的 IDE 调试.....	73
6.4 预处理.....	75
第 7 章 指针和函数.....	80
7.1 指针的基本概念.....	80

7.1.1 指针——地址.....	80
7.1.2 指针的说明和操作.....	81
7.2 有关指针的几个运算.....	82
7.3 关于函数的复习.....	85
7.4 函数的原型.....	88
7.4.1 函数原型示例.....	88
7.4.2 再谈函数的返回值.....	88
7.4.3 匹配.....	89
7.5 指针变量作为函数的参数.....	89
7.6 递归.....	93
* 第8章 关于函数的进一步讨论.....	101
8.1 返回指针.....	101
8.2 main()的返回值和参数.....	102
8.2.1 main()的返回值.....	102
8.2.2 main()的参数.....	103
8.3 值调用和引用调用.....	104
8.3.1 引用调用.....	104
8.3.2 从值调用到引用调用.....	104
8.4 函数指针.....	106
△第9章 图形子系统.....	112
9.1 有关图形的基本概念.....	112
9.2 图形方式示例.....	114
9.3 关于颜色.....	114
9.4 文本方式和图形方式.....	116
9.4.1 文本方式.....	116
9.4.2 图形方式.....	116
9.4.3 彩色.....	117
9.5 PC 中一个有趣的体系结构特点.....	117
9.6 图形应用举例.....	117
第10章 数据类型.....	122
10.1 结构的基本概念和用法.....	122
10.1.1 结构的概念.....	122
10.1.2 结构变量的赋值和引用.....	123
10.1.3 结构作为参数.....	123
10.2 结构数组和结构指针.....	124
10.2.1 结构数组.....	124
10.2.2 结构指针.....	127
10.2.3 结构内的数组和结构.....	128

10.3 枚举.....	128
第 11 章 关于数组、字符串和指针.....	135
11.1 数组和字符串的边界.....	135
11.1.1 数组的边界检查.....	135
11.1.2 字符串中的空字符.....	135
11.2 未定义长度的数组.....	136
11.3 开发一个游戏程序.....	136
11.3.1 游戏的规则.....	136
11.3.2 游戏的构思.....	137
11.4 指针操作的复习.....	137
11.4.1 指针是地址.....	137
11.4.2 类型十分重要.....	138
11.5 指针和数组.....	138
11.5.1 指针和下标.....	138
11.5.2 指针的下标.....	139
11.6 指针、数组和函数.....	139
11.6.1 向函数传递数组元素.....	139
11.6.2 传递整个数组.....	139
11.7 指针的指针和其他.....	140
11.7.1 指向指针的指针.....	140
11.7.2 指针的初始化.....	141
11.8 指针数组应用举例.....	141
第 12 章 位域、联合及高级操作符.....	150
12.1 用户自定义类型——位域.....	150
12.1.1 利用位域存储多个布尔量.....	150
12.1.2 使用位域示例.....	151
12.1.3 位域的其他特性.....	151
12.2 用户自定义类——联合.....	152
12.2.1 用联合查看接口信息.....	152
12.2.2 将 ASCII 字符按二进制显示.....	153
12.3 sizeof 和 typedef.....	154
12.3.1 sizeof.....	154
12.3.2 typedef.....	155
12.4 高级操作符.....	155
12.4.1 按位操作符“&”、“ ”、“~”和“^”.....	155
12.4.2 按位操作符“>>”和“<<”.....	157
12.4.3 利用操作符“<<”完成循环左移.....	157
12.4.4 “?”操作符.....	158

12.4.5 “,” 操作符.....	159
第 13 章 输入/输出和文件.....	163
13.1 格式化的控制台 I/O.....	163
13.1.1 printf 函数.....	163
13.1.2 printf 函数的格式命令.....	163
13.1.3 格式命令的修饰符.....	163
13.1.4 scanf 函数.....	164
13.1.5 scanf 函数的输入控制.....	164
13.2 ANSI I/O 系统.....	165
13.2.1 fread、fwrite、ferror、rewind 和 fseek 函数.....	165
13.2.2 打开文件的操作.....	167
13.2.3 读写字符和读写字符串.....	168
第 14 章 C 语言中的预处理.....	171
14.1 预处理的概念.....	171
14.2 #define、#include、#error 和#line.....	171
14.2.1 #define.....	171
14.2.2 #include.....	172
14.2.3 #error.....	172
14.2.4 #line.....	172
14.3 #if、#dse、#elif 和#endif.....	172
14.3.1 #if、#else 和#endif.....	172
14.3.2 #elif.....	173
14.4 #ifdef、#ifndef 和#undef.....	173
14.4.1 #ifdef.....	173
14.4.2 #ifndef.....	173
14.4.3 #undef.....	174
14.5 #pragma.....	174
14.5.1 startup 和 exit.....	174
14.5.2 inline 和 saveregs.....	175
14.5.3 warn、option 和 argused.....	175
14.5.4 hdrfile 和 hdrstop.....	175
△14.6 预定义宏名.....	175
* 第 15 章 C 语言的高级课题.....	177
15.1 C 语言的存储模式.....	177
15.1.1 存储模式.....	177
15.1.2 模式选择.....	178
△15.2 混合语言编程.....	178
15.2.1 混合语言编程的程序接口.....	178

15.2.2	混合编程的约定.....	179
15.2.3	编译与连接.....	179
15.2.4	C 语言对汇编语言的调用.....	179
15.3	鼠标的程序设计.....	182
15.3.1	几种常用的 Mouse 功能调用.....	182
15.3.2	一个控制鼠标的功能函数.....	184
15.4	用 C 语言实现简单的动画.....	184
15.4.1	符号动画.....	185
15.4.2	动画系统的高层菜单.....	185
15.4.3	调用过程中的几个功能函数.....	187
15.4.4	加载一个符号模块.....	189
15.4.5	动画显示系统.....	190
附录 A	快捷键及其意义.....	192
附录 B	转义字符表.....	193
附录 C	C 语言的关键字.....	193
附录 D	部分程序示例 1.....	193
附录 E	部分程序示例 2.....	194
附录 F	部分程序示例 3.....	195
附录 G	部分程序示例 4.....	196
参考文献		198

第 1 章 绪 论

任何事物之间的交流都需要一种工具，比如人与人之间的交流需要语言，语言就是表达对方的意图、思想的一种符号，正是语言的产生使人类的各种活动得以进行，语言在人们实施活动的过程中起到了一个媒介、桥梁的作用。人与计算机之间的交流也需要语言，我们把这种语言称为计算机语言，通过计算机语言表达人们的思想和意图，计算机接收语言信息，执行人们想要实现的操作，并将结果反馈给人们，如图 1-1 所示。

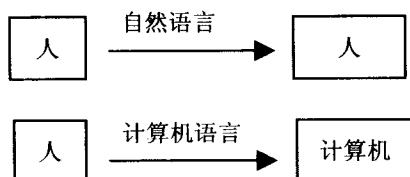


图 1-1 信息交换图

计算机是由电子元件构成的，它能直接识别的是电流的断通，用符号表示出来就是 0、1 代码，这也就是通常所说机器语言。由于计算机只识别 0、1 代码，因此应用对象主要是计算机专业人员，对于非专业人员，操作起来十分困难，比如难记忆、难书写。为了方便非专业人员对计算机的应用，扩大计算机的使用群体范围，产生了计算机的低级语言和高级语言。低级语言就是通常所说的汇编语言，高级语言主要有 Basic、Pascal、C、C++，还有可视化编程语言，如 Visual Basic、Visual C++、Visual FoxPro 等，计算机有这么多种语言，正如不同的国家讲不同的语言一样，它们都是在不同的环境下产生出来的，计算机高级语言也是在一中遵循具体的语法规则的特定环境下的计算机语言，是不同的开发商为了实现特定的功能而开发出来的一种计算机语言。图 1-2 为计算机语言的层次图。

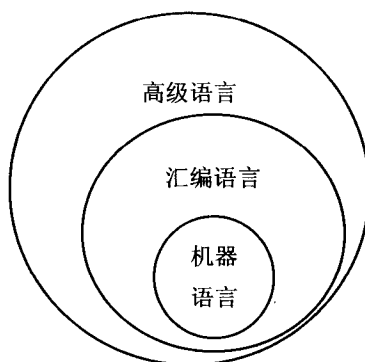


图 1-2 计算机语言的层次图

C 语言是国际上广泛流行的计算机高级语言。它适合于作为系统描述语言，既可用于编写系统软件，也可用于编写应用软件。

以前的操作系统等系统软件主要是用汇编语言编写的。汇编语言依赖于计算机硬件，程序的可读性和可移植性都比较差。为了提高可读性和可移植性，最好使用高级语言编写软件，

但一般的高级语言难以实现汇编语言的某些功能（汇编语言可以直接对硬件进行操作，例如对内存地址的操作、位操作等）。人们试图找到一种既具有一般高级语言的特性，又具有低级语言的特性的语言，集它们的优点于一身，于是，C 语言就在这种情况下应运而生了。

本章将对 C 语言的一般情况做简要介绍。为了有效地使用 C 语言，有一些相关的概念是必须了解或掌握的，1.4 节就来介绍这些概念。

1.1 C 语言概述

C 语言是由 DEC 公司的两位计算机专家 Ritchie 和 Thompson 于 20 世纪 70 年代末期发明的。尽管一开始 C 是在 UNIX 操作系统下开发成功的，但由于 C 语言的两个重要特性——高效率和可移植性¹，使它在极短的时间内便风靡全世界，两位 C 语言的发明者也成了计算机界的传奇人物。

C 语言是一种结构化语言。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，还可以直接访问内存的物理地址，进行位（bit）一级的操作。由于 C 语言实现了对硬件的编程操作，因此它集高级语言和低级语言的功能于一体，既可用于系统软件的开发，也适合于应用软件的开发。此外，C 语言还具有效率高、可移植性强等特点。因此广泛地移植到了各类计算机上，从而形成了多种版本的 C 语言。

C 语言除了上述两条最为基本的优良特性外，还有不少优点，一般认为以下两点是 C 语言受到众多程序员青睐的重要原因。一是用 C 语言可以写出清晰、简洁、风格优美的程序；二是 C 语言对于程序员很少有限制，程序员在使用 C 语言时可以“随心所欲”，而不像使用绝大多数高级语言时，程序员觉得与计算机隔了一段令人不快的距离²。

目前，许多程序员在各个领域内使用 C，事实上，程序员们倾向于把 C 看成一种通用的工具，可以在任何领域内使用。一般认为 C 语言比较擅长的领域有：操作系统、编译系统、数据库系统、计算机网络，还有某些实时应用等。

1.2 C 和 C++

C 语言是 C++语言的基础，C++语言和 C 语言在很多方面是兼容的，因此，掌握了 C 语言，在进一步学习 C++时就能以一种熟悉的语法来学习面向对象的语言，从而达到事半功倍的目的。目前最流行的 C 语言有以下几种版本。

- Microsoft C，简称为 MSC。
- Borland Turbo C，简称为 Turbo C。
- QUICK C。

这些 C 语言的版本不仅实现了 ANSI C 标准，而且在此基础上各自做了一些扩充，使其更加方便、完美。

学习 C，很有必要了解 C++，如果学习者以后要学习 C++，一般总是要先学习 C 作为其基

¹ 可移植性是指某个程序可以运行在不同的计算机上。可移植性有级别的区分，一般所说的都是指源程序的可移植性。比如 UNIX 系统上的某个 C 源程序，只要将其传送到微机上，用微机上的 C 语言工具（比如 Turbo C++ 2.0）编译连接，就可以在微机上运行。

² 程序员往往都以对计算机的熟练运用为骄傲，高级语言一般并不假定使用者对于计算机了如指掌，因此高级语言在允许做什么和不允许做什么之间有相当明确的界限。C 语言则不同，它假设使用者就是程序员，因此基本上不存在什么限制。

础；如果学习者不学习 C++，了解 C++ 对于学习 C 也是会有好处的。从名称上来说，C++ 显然和 C 有着密切的关联，通常人们比较强调这一方面的联系，比如 C 程序可以在 C++ 环境中正确执行，而 C++ 程序和 C 程序相似，也具有十分高的效率等。但是从设计思想上来看，C++ 有更多的与 C 不相似的地方。可以这样说，C++ 从 C 语言那里继承了相当一部分技术优势，C++ 利用这些技术为一个全新的思想——面向对象（面向对象这一思想对于 C 来说是全然陌生的）。

那么什么是面向对象呢？简要地说，面向对象有两方面的意义：对于实际编程人员来说，面向对象的方法提供了一种框架，利用此框架，他们可以更加方便地理解、规划、管理以及开发庞大的项目；从理论上来说，面向对象是这样一种体系（和方法），程序员可以更加真实地描写实际世界中的客体。具体来说，面向对象的观点认为描述一个客体仅用各个量来说明（比如长、宽、重和颜色等）是不够的，还应该说明客体的行为（比如旋转、放大、存储和删除等）。简言之，面向对象的思想更接近于人类理解世界的方式，从这一点上来看，面向对象有着极大的潜力和发展前景。

1.3 为什么学习 C

考虑到 C 语言有这么多优点，回答这一问题似乎有点多余。如果一定要回答，人们通常给出两种理由：C 是最流行的计算机语言，而且 C 语言的用途十分广泛（是一种万能的工具，当然应该学）。事实上重要的问题不是 C 语言是否值得学习，而是学习者希望通过学习 C 语言达到什么样的目的。

1.4 几个基本概念

对于初学者，可能尚未接触过本节提到的基本概念，但这些概念却是比较重要的。

1. 编译和解释

计算机语言（尤其是高级语言）是为了程序员开发方便而设计的，为了让计算机执行，必须通过编译或解释的方法。编译指的是将程序一次性转换成计算机可以识别的二进制码的集合；解释是指一边转换一边执行。早期的 BASIC 和 LISP 就采用解释的方法，更多的语言采用编译的方法，另外不少语言同时提供上述两种方式。

2. 源文件、目标文件和可执行文件

源文件就是程序员所写的程序，源文件编译后就产生目标文件，目标文件经过连接³后产生可执行文件。

3. 操作系统

任何可执行文件都必须取得操作系统的支持才能运行。对于微型机来说，操作系统一般是指 DOS 和 Windows。早年的 C 和 UNIX 操作系统有着密切的联系，但标准化的 C 不依附于任何一种操作系统，而且得到目前几乎所有操作系统的支持。在这种意义上，C 语言是计算机界的“世界语”。现在的 C 语言更多的是在 Windows 环境下执行。

³ 连接是为了将多个目标文件合为一个可执行文件而预备的，单个目标文件只是多个目标文件的一个特例，在形成可执行文件时也需要连接的操作。

4. 低级语言、高级语言和中级语言

低级语言指的是汇编语言，由于汇编指令和机器码有一一对应的关系，因此，除去那些有助于编程的功能以外，汇编就是机器指令。一般来说，如果对于计算机的硬件十分熟悉，那么使用汇编语言就是理所当然的。高级语言不太注重计算机的个性，高级语言都带有某种抽象的特性，正是由于这种抽象，不同的高级语言往往在某些领域内比较擅长，比如 FORTRAN 用于科学计算，LISP 和 PROLOG 用于人工智能等。

C 语言有着自身的特殊性。一方面，C 语言有着高级语言的一般特性（有丰富的控制流，允许用户自定义数据类型，有方便的输入/输出等）；另一方面，C 语言又可以接触到计算机的底层（在 C 中可以直接写汇编指令，C 的执行效率仅次于汇编，高于所有其他高级语言；另外，用 C 编译出来的可执行文件比其他高级语言编译的文件要小。正因为这样，C 语言又被人们称为中级语言。

5. 数据结构和计算机算法

为了利用计算机解决实际问题，一般需要两个方向上的努力：一是提供一种工具（环境），利用它，人们可以方便地操作计算机；二是将人的理解（模型、定义、定理和规则等）转化为计算机可以接受的样式。第一种努力以计算机语言最为突出，而第二种努力则主要表现在数据结构和计算机算法上。数据结构指的是在实际世界中的客体（这些客体在计算机看来都是“数据”）中抽取出较为一般性的“结构”，而这些“结构”是计算机操作的基本对象。抽取结构之后的任务是对这些结构做运算。客体之间如何相互作用决定了如何运算，这一过程就是由计算机算法来处理的。C 语言在结构上有很大的自由度，很可能是因为这一点使 C 语言有十分宽广的应用前景。同时这一点对于学习者来说将是一个不小的困难，有效地利用 C 语言解决实际问题学习掌握 C 语言的最好途径。

△1.5 C++的特点

C++之所以成为一种流行的编程语言，是因为它保持了 C 语言的精髓，并有自己独到的特点，可以归纳为：

- (1) 语言表达能力强⁴。
- (2) C++简洁、紧凑，使用方便、灵活，易于学习和应用。
- (3) 数据类型丰富，具有现代语言的各种数据结构（读者可以想想以前学过的其他高级语言中的数据结构⁵）。
- (4) C++生成的代码质量高、可移植性好。
- (5) 使软件的可重用性、可扩充性、可维护性和可靠性有了很大提高。

⁴ C++是“面向对象程序设计”的语言，提供了控制和管理对象的机制，也就是说，它使程序结构清晰、易于扩展、易于维护，也使解决问题的机制更符合客观实际。

⁵ Borland C++除了与早期的版本的 Turbo C, Turbo C++兼容外，还有很多自己的特色，Borland C++包含一定的扩展，使程序员能够用混合语言和混合模型编程来扩充 PC 的能力。为了帮助用户入门，C++还提供了一些类库，还可以编制 Windows 应用程序，提供了预编译的头文件，提供了实模式和保护模式编译器。类库中提供了集合、数组等类型。另外，无论是在实模式还是在保护模式，它都支持鼠标、多重窗口、多重事件编辑器、内部汇编码、嵌入式汇编器和集成调试器等。C++也有 9 个语句，与 C 语言相同，并在 C 语言的基础上增加了 17 个关键字，其中 catch、try、throw 和 template 是试验性的，供将来使用。

第2章 Turbo C 开发环境

Turbo C 是一个集成开发环境,集成指的是用户可以在 Turbo C 的环境下编写、修改、保存、编译、执行并调试 C 和 C++ 的程序。本章将从 3 个方面对该环境做介绍,这 3 个方面是窗口环境、开发环境和编辑环境。

2.1 Turbo C 概述

Turbo C++ 是在 Turbo C 的基础上发展而来的。Turbo C 的程序在 Turbo C++ 的环境下照样能够运行,而 Turbo C++ 在应用方面更是取得了丰硕的成果。Turbo C++ 能获得如此成功,主要原因有两条:一是面向对象的方法趋于成熟,而且正被越来越多的人接受;二是 Borland 公司成功地将理论付诸于实践。另外,兼容是又一个不容忽视的因素,C++ 在保持向 C 兼容方面取得了完满的结果。所以 C++ 与我们所要学习的 C 语言并不矛盾,掌握得快的读者可学习更高级的 C++。

1. C 源程序的结构特点

- (1) 一个 C 语言源程序可以由一个或多个源文件组成。
- (2) 每个源文件可由一个或多个函数组成。
- (3) 一个源程序不论由多少个文件组成,都有一个且只能有一个 main 函数,即主函数。
- (4) 源程序中可以有预处理命令(include 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。
- (5) 每一个说明、每一个语句都必须以分号结尾,但对于预处理命令、函数头和花括号“}”之后不能加分号。
- (6) 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,也可不再加空格来间隔。

2. 书写程序时应遵循的规则

从书写清晰,便于阅读、理解和维护的角度出发,在书写程序时应遵循以下规则。

- (1) 一个说明或一个语句占一行。
- (2) 用“{”括起来的部分通常表示了程序的某一层结构。“{”一般与该结构语句的第 1 个字母对齐,并单独占一行。
- (3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,以便看起来更加清晰,增加程序的可读性。在编程时应力求遵循这些规则,以养成良好的编程风格。

3. C 语言的字符集

字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

- (1) 字母。大写字母 A~Z 共 26 个,小写字母 a~z 共 26 个。
- (2) 数字。0~9 共 10 个。
- (3) 空白符。空格符、制表符和换行符等统称为空白符。空白符只在字符常量和字符

串常量中起作用，在其他地方出现时，只起间隔作用，编译程序对它们忽略。因此在程序中是否使用空白符对程序的编译不产生影响，但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

(4) 标点和特殊字符。

4. C 语言的词汇

在 C 语言中使用的词汇分为 6 类：标识符、关键字、运算符、分隔符、常量和注释符。

(1) 标识符。在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。C 规定，标识符只能是字母 (A~Z, a~z)、数字 (0~9) 和下划线 (__) 组成的字符串，并且其第 1 个字符必须是字母或下划线。

以下标识符是合法的：a、x、_3x、BOOK_1、sum5。

以下标识符是非法的：3s (以数字开头)、s*T (出现非法字符 “*”)、-3x (以减号开头)、bowy-1 (出现非法字符 “-”)。

在使用标识符时还必须注意以下几点。

①标准 C 不限制标识符的长度，但它受各种版本的 C 语言编译系统的限制，同时也受到具体计算机的限制。例如在某版本的 C 中规定标识符的前 8 位有效，当两个标识符的前 8 位相同时，被认为是同一个标识符。

②在标识符中，大小写是有区别的。例如 BOOK 和 book 是两个不同的标识符。

③标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号，因此，命名应尽量有相应的意义，以便阅读和理解，做到“顾名思义”。

(2) 关键字。关键字是由 C 语言规定的具有特定意义的字符串，通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类。

①类型说明符。用于定义、说明变量、函数或其他数据结构的类型。如前面例题中用到的 int、double 等。

②语句定义符。用于表示一个语句的功能。如例 1.3 中用到的 if else 就是条件语句的语句定义符。

③预处理命令字。用于表示一个预处理命令。如前面各例中用到的 include。

(3) 运算符。C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。

(4) 分隔符。在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中，分隔各个变量；空格多用于语句的各单词之间，作为间隔符。在关键字、标识符之间必须要有一个以上的空格符作为间隔，否则将会出现语法错误，例如把 “int a;” 写成 “inta;”，C 编译器会把 inta 当成一个标识符处理，其结果必然出错。

(5) 常量。C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在第 2 章中将专门予以介绍。

(6) 注释符。C 语言的注释符是以 “/*” 开头并以 “*/” 结尾的串，在 “/*” 和 “*/” 之间的即为注释。程序编译时，不对注释做任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，

使翻译跳过不做处理，等调试结束后再去掉注释符。

为了说明 C 语言源程序结构的特点，先看以下两个程序。这几个程序由简到难，表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中了解到组成一个 C 源程序的基本部分和书写格式。

【例 2.1】

```
main()
{
    printf("C 语言世界 www.vcok.com, 您好! \n");
}
```

main 是主函数的函数名，表示这是一个主函数。每一个 C 源程序都必须有且只能有一个主函数（**main** 函数）。**printf** 函数的功能是把要输出的内容送到显示器去显示。**printf** 函数是一个由系统定义的标准函数，可在程序中直接调用。

【例 2.2】

```
#include "stdio.h"
#include "math.h" /*include 称为文件包含命令，扩展名为 .h 的文件称为头文件或首部文件，定义两个实数变量，以备后面程序使用*/
main()
{
    double x,s; /*定义两个实数变量，以备后面程序使用*/
    printf("input number:\n"); /*显示提示信息*/
    scanf("%lf",&x); /*从键盘获得一个实数 x*/
    s=sin(x); /*求 x 的正弦，并把它赋给变量 s*/
    printf("sine of %lf is %lf\n",x,s); /*显示程序的运算结果*/
} /*main 函数结束*/
```

程序的功能是从键盘输入一个数 x ，求 x 的正弦值，然后输出结果。在 **main()** 之前的两行称为预处理命令。预处理命令还有其他几种，这里的 **include** 称为文件包含命令，其意义是把尖括号“<>”或引号“”内指定的文件包含到本程序中，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为 **.h**，因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此，只要是在程序中调用一个库函数，都必须包含该函数原型所在的头文件。在本例中，使用了 3 个库函数：输入函数 **scanf**、正弦函数 **sin** 和输出函数 **printf**。**sin** 函数是数学函数，其头文件为 **math.h** 文件，因此在程序的主函数前用 **include** 命令包含了 **math.h**。**scanf** 和 **printf** 是标准输入/输出函数，其头文件为 **stdio.h**，在主函数前也用 **include** 命令包含了 **stdio.h** 文件。

需要说明的是，C 语言规定对 **scanf** 和 **printf** 这两个函数可以省去对其头文件的包含命令，所以在本例中也可以删去第 2 行的包含命令“**#include**”。同样，在例 2.1 中使用了 **printf** 函数，也省略了包含命令。

在例题中的主函数体中又分为两部分，一部分为说明部分，另一部分为执行部分。说明是指变量的类型说明。例题中未使用任何变量，因此无说明部分。C 语言规定，源程序中所有用到的变量都必须先说明、后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，与解释型的 BASIC 语言是不同的。说明部分是 C 源程序结构中很重要的组成部