策划与组编：湖北省计算机学会·诺亚教科文中心

新世纪高等学校计算机系列教材

C 程序设计基础

主　编　黄远林
编　者　黄远林　张冬梅　范玉莲

高等教育出版社·北京
中山大学出版社·广州

图书在版编目（CIP）数据

C 程序设计基础 /黄远林 主编. —北京: 高等教育出版社, —广州: 中山大学出版社. 2004.1
（新世纪高等学校计算机系列教材/湖北省计算机学会，诺亚教科文中心 组编）
ISBN 7-306-02210-5

Ⅰ.C … Ⅱ. 黄 .. Ⅲ. 电子计算机-C 语言-高等学校-教材 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2003）第 110708 号

内 容 简 介

本书以标准 C 为框架，以 Torbo C 为背景，按照紧扣基础、面向应用、强调算法及兼顾等级考试的撰写原则，较全面地介绍了用 C 语言进行程序设计的基本思想、方法和技术。并且，在侧重程序设计实践介绍的同时，对 C 语言的表达式、类型转换、变量的存储和函数、指针等重点及难点内容，也结合大量应用示例分别作了较为明确而又详细的介绍。

本书概念清楚、语言精练、叙述清晰、内容全面、例习题丰富，书中所有示例程序均上机调试运行和给出了结果，且大都给出了对算法思路的分析及算法步骤，因而很适于教学和初学者自学训练使用。

本书适合高等院校各类专业作为教材使用，亦适合各行各业的软件开发人员与广大欲参加计算机等级考试的人员参考。

版权所有 盗印必究

C 程序设计基础 © 黄远林 主编

责任编辑：里 引 唐 源 封面设计：袁 作
责任校对：诺 亚 责任技编：潘 隆

出版发行：高等教育出版社 （地址：北京市东城区沙滩后街 55 号 邮编：100009）
中山大学出版社 （地址：广州市新港西路 135 号 邮编：510275）

经 销：广东新华发行集团股份有限公司
武汉诺亚信息传播有限公司 （电话：027—87597347（带传真） 87596532 邮编：430073）

印 刷：华中理工大学印刷厂

开 本：787mm×1092mm 1/16 印 张：20 字 数：500 千字
版 次：2004 年 1 月第 1 版 印 次：2004 年 1 月第 1 次印刷 印 数：1— 7 000

定 价：26.80 元

新世纪高等学校计算机系列教材
编审指导委员会

主　　任：卢正鼎（华中科技大学教授、博士生导师）

副 主 任：何炎祥（武汉大学教授、博士生导师）

编　　委：（以姓氏笔画排序）

王元珍　（华中科技大学教授、博士生导师）

叶俊民　（华中师范大学副教授、博士）

李　兵　（湖北大学副教授、博士）

李鸣山　（武汉大学教授）

陈　珉　（武汉大学教授、博士）

陈传波　（华中科技大学教授、博士生导师）

陈建勋　（武汉科技大学教授、博士）

陆际光　（中南民族大学教授）

汪厚祥　（海军工程大学教授、博士）

程元斌　（江汉大学副教授）

程学先　（湖北工学院教授）

谭连生　（华中师范大学教授、博士）

熊家军　（空军雷达学院教授、博士）

戴光明　（中国地质大学教授、博士）

执行编委：唐元瑜　（华中科技大学副编审）

余健棠　（华中科技大学编审）

此为试读，需要完整PDF请访问：www.ertongbook.com

新世纪高等学校计算机系列教材

总　　序

21世纪人类已跨入了信息时代，以计算机为核心的信息技术正在迅猛发展，并不断改变着人类社会的工作方式、生产方式、生活方式和学习方式。当今，各行各业的现代化都离不开计算机，各行各业的人们都在学习和使用计算机，而计算机科学技术及其教育本身也在日新月异地发展变化。为了顺应时代的潮流，满足新世纪高等学校计算机教育事业发展、教学改革和人才培养对高质量特色教材的需求，湖北省计算机学会及其教育与培训专业委员会和诺亚教科文中心等共同策划、组织并约请华中科技大学、武汉大学、华中师范大学、中国地质大学、中南民族大学、武汉科技大学、海军工程大学、空军雷达学院、湖北大学、湖北工学院、江汉大学等高校长期奋斗在教学科研第一线，且具有丰富教学实践经验的部分优秀骨干教师共同编写了这套《新世纪高等学校计算机系列教材》。

这套系列教材共40余种，主要是根据中国计算机学会教育委员会、全国高等学校计算机教育研究会等联合推出的《中国计算机科学与技术学科教程2002》（简称“CCC2002教程”）中的课程体系与课程大纲的要求，进行规划和组织编写的，并主要供高等学校计算机科学与技术专业本科教学使用。对于本系列教材中部分供软件学院等使用的教材，大都是一些理论与实践并重，从内容到结构都是全新的特色教材，并且有些教材也可适合其他专业选用。

当今，计算机科学技术突飞猛进地向前发展，计算机新技术和新产品不断涌现，高等教育事业和教学改革不断深化，国内教育逐步与国际教育接轨，社会对计算机专业人才的要求越来越高，等等。面对这些新形势，这套系列教材以培养学生具有较扎实的专业基础理论知识、实践能力、创新能力和较高的综合素质能力为目的，既注重知识的更新与合理的结构，又注意学习和吸取国内外优秀教材的优点与精华，并尽力反映国内外最新的教学科研成果及作者们宝贵的实践经验。

我相信，通过作者们的共同努力，定能将这套系列教材打造成为一套既具有时代特色，又非常适用的、高质量的系列教材，为我国高等教育事业的发展和高素质专业人才的培养作出应有的贡献。

新世纪高等学校计算机系列教材
编审指导委员会主任

卢正鼎

2003年7月

前　言

人类已步入信息化的21世纪，随着社会经济向知识经济的发展，在科学、技术、经济、文化和军事等各个领域都需要高素质的计算机人才。为了促进21世纪计算机教育的改革和发展，满足高等学校为培养高素质、复合型人才对高质量特色教材的需求，我们编写了此书。

C语言是近年来在国内外被广泛学习、普遍应用的一种计算机程序设计语言。C语言具备一切高级语言的特征和优势，既可清晰地体现结构化、模块化程序设计的思想，又具有低级语言的许多特点和精华，例如可直接访问内存地址，对字节的位进行运算等，从而可大大提高程序的效率。C语言功能多、表达能力强、使用灵活方便，它支持自顶向下逐步求精的程序设计技术，其函数式结构为实现程序的模块化设计提供了强有力的保障，因此，C语言被广泛应用于各类系统软件和应用软件的开发。

C语言语法规则多、概念复杂、用法灵活，初学者难于理解和掌握，因而学习难、深入难是广大初学者的共同感受。针对上述问题，本书按照紧扣基础、面向应用、强调算法的撰写原则，力图将C语言复杂的概念通俗化，内容及文字叙述条理化，在使之适合初学者学习和掌握的前提下，以标准C为基本框架，以Turbo C为背景，较全面地介绍了使用C语言进行程序设计的基本思想、方法和技术，为读者们今后进一步提高程序设计的能力奠定基础，特别是为后继学习Windows环境下的程序设计，为学习面向对象的程序设计打下坚实的基础。

全书共分12章，其内容包括：第1章介绍了C语言概述；第2章介绍了算法；第3章介绍了数据类型及其运算；第4~6章分别介绍了顺序、选择、循环结构程序设计；第7~9章分别介绍介绍了数组、函数和指针；第10~12章分别介绍了预处理、结构体和共用体、文件。本书中的例题均已通过上机调试并附有运行结果。

本书的特点是：循序渐进，由浅入深，通俗易懂，基本概念、基本原理与基本应用的介绍与应用层次清晰，脉络分明，而且所有例题大都有算法分析，重点突出，分散难点，精选习题，兼顾等级考试。

本书是编者们在总结多年从事C语言研究应用和教学与实践经验基础上，参考了国内外有关资料而编写的，既可作为大专院校计算机专业或非计算机专业（包括成教、职业技术学院）的教材使用，又可供编程人员、自学人员们自学参考。

本书由武汉科技大学黄远林副教授主编，具体编写人员有：黄远林（第3、8、9、10章及附录），张冬梅（第1、7、11、12章），范玉莲（第2、4、5、6章及3.2.7节）。本书在编写过程中，得到了湖北省计算机学会及其教育与培训专业委员会、武汉科技大学、中国地质大学以及新世纪高等学校计算机系列教材编审指导委员会等有关领导与专家的大力支持与帮助，在此一并致谢。

由于编者水平有限，书中缺点和错误在所难免，敬请专家和读者批评批正。

编　者

2003年10月

目　　录

第 1 章　C 语言概述 …… (1)

1.1　程序设计及程序设计语言 …… (1)

1.2　C 语言出现的历史背景 …… (3)

1.3　C 语言的特点 …… (4)

1.4　简单的 C 程序举例 …… (5)

1.5　C 程序的上机步骤 …… (8)

习题一 …… (13)

第 2 章　算法 …… (14)

2.1　算法的概念 …… (14)

2.2　算法的表示及其特性 …… (15)

2.2.1　算法的表示方法 …… (15)

2.2.2　算法的特性 …… (19)

2.3　算法设计和描述举例 …… (20)

习题二 …… (22)

第 3 章　数据类型、运算符与表达式 …… (24)

3.1　C 语言的数据类型 …… (24)

3.1.1　C 语言的数据类型简介 …… (24)

3.1.2　常量与变量 …… (25)

3.1.3　整型数据 …… (26)

3.1.4　实型数据 …… (31)

3.1.5　字符型数据 …… (33)

3.1.6　变量赋初值 …… (37)

3.1.7　各类数值型数据之间的混合运算 …… (37)

3.2　C 语言的运算符与表达式 …… (40)

3.2.1　C 语言的运算符简介 …… (40)

3.2.2　算术运算符表达式 …… (40)

3.2.3　赋值运算符和赋值表达式 …… (43)

3.2.4　逗号运算符和逗号表达式 …… (46)

3.2.5　关系运算符和关系表达式 …… (47)

3.2.6　逻辑运算符和逻辑表达式 …… (49)

3.2.7　位运算符和位运算表达式 …… (51)

习题三 …… (53)

第 4 章　顺序结构程序设计 …… (55)
4.1　表达式语句 …… (55)
4.2　基本的标准输入输出函数 …… (56)
4.2.1　字符输入和输出函数 (getchar, putchar) …… (56)
4.2.2　格式输出函数和格式输入函数(printf ,scanf) …… (57)
4.3　顺序结构程序设计举例 …… (61)
习题四 …… (64)
第 5 章　选择结构程序设计 …… (65)
5.1　复合语句 …… (65)
5.2　if 语句 …… (66)
5.2.1　if 语句的两种基本形式 …… (66)
5.2.2　if 语句的嵌套 …… (68)
5.3　switch 语句 …… (73)
习题五 …… (77)
第 6 章　循环结构程序设计 …… (78)
6.1　while 语句 …… (78)
6.2　do—while 语句 …… (80)
6.3　for 语句 …… (83)
6.4　循环的嵌套 …… (86)
6.5　几种循环语句使用小结 …… (89)
6.6　break 语句与 continue 语句 …… (90)
6.6.1　break 语句 …… (90)
6.6.2　continue 语句 …… (91)
6.7　goto 语句和带标号语句 …… (93)
6.7.1　goto 语句 …… (93)
6.7.2　带标号语句 …… (93)
习题六 …… (95)
第 7 章　数组 …… (98)
7.1　数据结构与数组的概念 …… (98)
7.2　一维数组的定义和引用 …… (99)
7.3　二维数组的定义和引用 …… (105)
7.4　字符串与字符数组 …… (111)
习题七 …… (118)
第 8 章　函数 …… (120)
8.1　函数定义的一般形式 …… (120)
8.2　函数的参数和函数的值 …… (123)
8.2.1　形式参数和实际参数 …… (123)

此为试读,需要完整PDF请访问: www.ertongbook.com

8.2.2 函数的返回值 …… (126)
8.3 函数的调用 …… (129)
8.3.1 函数调用的一般形式 …… (129)
8.3.2 函数调用的方式 …… (131)
8.3.3 被调用函数的声明和函数原型 …… (136)
8.4 函数的嵌套调用 …… (139)
8.5 函数的递归调用 …… (141)
8.6 数组作为函数参数 …… (147)
8.6.1 数组元素作函数实参 …… (147)
8.6.2 数组名作为函数参数 …… (148)
8.7 局部变量和全局变量 …… (157)
8.7.1 局部变量 …… (157)
8.7.2 全局变量 …… (159)
8.8 变量的存储类别 …… (162)
8.8.1 变量的动态与静态存储方式 …… (162)
8.8.2 动态存储的 auto 自动类局部变量 …… (162)
8.8.3 用 static 声明的静态存储类局部变量 …… (163)
8.8.4 register 寄存器变量 …… (164)
8.8.5 用 extern 声明外部变量 …… (165)
8.8.6 内部函数和外部函数 …… (166)
习题八 …… (167)
第 9 章 预处理命令 …… (172)
9.1 宏定义 …… (172)
9.1.1 无参宏定义 …… (172)
9.1.2 带参宏定义 …… (175)
9.2 文件包含 …… (180)
9.3 条件编译 …… (180)
习题九 …… (183)
第 10 章 指针 …… (184)
10.1 地址和指针的基本概念 …… (184)
10.2.1 概述 …… (186)
10.2.2 指针变量的定义 …… (188)
10.2.3 指针变量的赋值和引用 …… (189)
10.2.4 指针变量作为函数参数 …… (198)
10.2.5 有关指针变量几个问题的进一步说明 …… (201)
10.3 数组指针和指向数组的指针变量 …… (205)
10.3.1 指向数组元素的指针 …… (205)
10.3.2 通过指针访问一维数组 …… (206)

10.3.3 通过指针在函数间传递一维数组 …… (208)
10.3.4 指向多维数组的指针和指针变量 …… (215)
10.3.5 通过指针在函数间传递多维数组 …… (218)
10.4 指针与字符串 …… (222)
10.4.1 通过指针访问字符串 …… (222)
10.4.2 使用字符串指针变量与使用字符数组的区别 …… (232)
10.5 函数指针变量 …… (233)
10.6 指针型函数 …… (237)
10.7 指针数组和指向指针的指针 …… (239)
10.7.1 指针数组的概念 …… (239)
10.7.2 指向指针的指针 …… (244)
10.7.3 main函数的参数 …… (246)
10.8 有关指针的数据类型和指针运算的小结 …… (247)
习题十 …… (248)
第11章 结构体与共用体 …… (250)
11.1 概述 …… (250)
11.2 结构体与结构体类型变量 …… (251)
11.2.1 结构体类型的定义 …… (251)
11.2.2 结构体类型变量的定义 …… (252)
11.2.3 结构体类型变量的引用 …… (254)
11.2.4 结构体类型变量的初始化 …… (256)
11.3 结构体数组 …… (257)
11.4 指向结构体类型数据的指针及其使用 …… (260)
11.5 用指向结构体类型数据的指针处理链表 …… (266)
11.6 共用体 …… (276)
习题十一 …… (281)
第12章 文件 …… (282)
12.1 文件概述 …… (282)
12.2 文件类型指针 …… (284)
12.3 文件的各种操作 …… (284)
12.4 文件的随机读写操作 …… (295)
12.5 文件操作中出错的检测 …… (298)
习题十二 …… (298)
附录 …… (300)
附录A ASCII代码与字符对照表 …… (300)
附录B C语言的运算符及其优先级和结合性 …… (301)
附录C C库函数 …… (302)
参考文献 …… (308)

C语言概述

C语言是目前世界上流行和使用最为广泛的一种程序设计语言。本章在简单程序设计、程序设计语言的基础上，介绍了C语言出现的历史背景及特点。并通过几个简单的C程序，介绍了C语言程序的结构和基本格式。通过这些内容的学习，将使读者对C语言程序设计有一个初步的了解，为以后各章的学习打下必要的基础。

1.1 程序设计及程序设计语言

人类日常用来交流的语言称为自然语言，如汉语、英语、俄语、日语等。但计算机并不能直接理解自然语言。这是由于计算机是机器，能够识别的只是电信号。电信号有两个，一个是高电平，一个是低电平。因此，计算机能够直接识别的只是表示电压有、无的0、1代码。要控制计算机工作，就必须将人的意图用计算机能够识别的方式表达出来。计算机语言就是计算机能够理解执行的语言。用计算机语言书写的一组有一定逻辑关系的指令就是程序。

计算机语言的种类很多，总的来说可以分成机器语言、汇编语言、高级语言三大类。

1. 第一代语言——机器语言

二进制数0和1是计算机的工作基础。机器语言就是用二进制数表示的机器指令。例如，A+B用机器语言可表示为：

1000110010100000

由于机器代码可以直接被计算机识别，因此机器语言最大的特点便是效率高、执行速度快。但是采用机器代码编写程序，要求程序员熟练地记住所有机器指令的二进制代码、数据单元地址和指令地址。这样，编程困难，工作量大，易于出错，而且编写的程序可读性极差。

2. 第二代语言——汇编语言

基于机器语言的种种缺点，人们开始寻找一种接近人类语言的表达方式。将机器代码采用英文缩写的助记符标识，这就是汇编语言。例如，A+B用汇编语言可表示为：

A ADD B

该命令再通过计算机的一些翻译软件(汇编程序)转换成0、1代码才能被计算机执行。汇编语言是与机器语言一一对应的一种程序设计语言。虽然它不是用0、1代码编写，但是实质和机器语言是相同的，都是直接对硬件操作，只不过指令采用助记符标识，更容易识别和记忆。汇编程序通常由三部分组成：指令、伪指令和宏指令。汇编程序的每一条指令只能对应实际操作过程中的一个很细微的动作，例如移动、自增，因此汇编源程序一般比较冗长、复杂、容易出错，而且使用汇编语言编程需要有更多的计算机专业知识。因此，人们只有在直接编写面向硬

件的驱动程序时才采用它。

机器语言和汇编语言均与特定的计算机硬件有关，属于低级语言。

3. 第三代语言——高级语言

高级语言起始于 20 世纪 50 年代中期，是一种能够被计算机接收同时更接近自然语言的程序设计语言。和汇编语言相比，高级语言不但将许多相关的机器指令合成为单条指令，并且去掉了与具体操作有关但与完成工作无关的细节，例如堆栈、寄存器的使用等，大大简化了程序中的指令。同时，由于省略了很多细节，编程者不需要有太多的计算机专业知识。因此，使用高级语言编写的程序可读性强，编程方便。如 BASIC(True Basic、QBasic、Virtual Basic)、C、PASCAL、FORTRAN、智能化语言(LISP、PROLOG)等都是高级语言。

人们用高级语言所编写的程序叫源程序。源程序不能直接被计算机识别，必须经过转换才能被执行。转换方式分为两类：

1) 解释方式

解释方式类似于日常生活中的“同声翻译”。一边将源代码由相应语言的解释器“翻译”成目标代码，一边执行，因此效率比较低，而且不能生成可独立执行的可执行文件。但这种翻译方式比较灵活，可以及时发现错误，动态地调整、修改应用程序。如 BASIC、PROLOG 语言采取的就是解释执行的方式。

2) 编译方式

编译方式是指在应用源程序执行之前，就将程序源代码“翻译”成目标代码(机器语言)，因此其目标程序可以脱离其语言环境独立执行，使用比较方便、效率较高。但应用程序一旦需要修改，必须先修改源代码，再重新编译生成新的目标文件(*.OBJ)才能执行。目前大多数的高级语言都是编译型的，如 PASCAL、FORTRAN、C 等。

目前高级语言的种类有近千种之多。常用的有 BASIC、COBOL、FORTRAN、PASCAL、C/C++ 和 JAVA 等语言。其中：

1) BASIC 语言

BASIC 语言是当前国内外广泛使用的、适合于初学者的一种计算机语言，它适用于数值计算、事务管理、绘画和游戏等，易学实用。几乎所有小型计算机和微型计算机都配置了 BASIC 语言。

2) FORTRAN 语言

FORTRAN 是第一个被广泛使用的高级程序设计语言，主要用于科学计算。

3) PASCAL 语言

PASCAL 语言是是世界上第一个结构化程序设计语言，由于 PASCAL 语言具有良好的结构化程序设计特性，适合于培养学生掌握自顶向下逐步求精的结构化程序设计思想和方法，因此国内外许多大学往往将 PASCAL 作为第一门程序教学语言。PASCAL 语言提供了丰富的数据类型和语句，功能强，应用广，编译和运行效率高。另外 PASCAL 语言标准化程度高，不依赖于具体的机器，可移植性好。

4) JAVA 语言

JAVA 语言是 Sun 公司推出的新一代面向对象的程序设计语言。具有简单、面向对象、分布式、解释、安全、结构中立、可移植性好、性能优异、多线程等特点，它的平台无关性特别适合在 Internet 环境上开发应用系统。使用 JAVA 语言可以制作大部分的网络应用程序系统。

此为试读，需要完整PDF请访问：www.ertongbook.com

1.2　C 语言出现的历史背景

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。是 20 世纪 70 年代美国贝尔(Bell)实验室为研制和开发 UNIX 操作系统而开发的一种系统描述语言。它既可用来编写系统软件，也可用来编写应用软件。

1969 年，贝尔实验室的 Ken Thompson 和 Dennis M. Ritchie 开始研制 UNIX 操作系统，用了不到两个人年的时间就研制成功了。UNIX 的早期版本是用汇编语言编写的。但由于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。为了提高程序的可读性和可移植性，人们希望改用高级语言编写系统软件，但一般的高级语言不能像汇编语言一样直接对硬件进行操作。有没有一种既具有一般高级语言特性，又具有低级语言特性的语言呢？在这种情况下 C 语言应运而生。图 1.1 显示了 C 语言的由来。

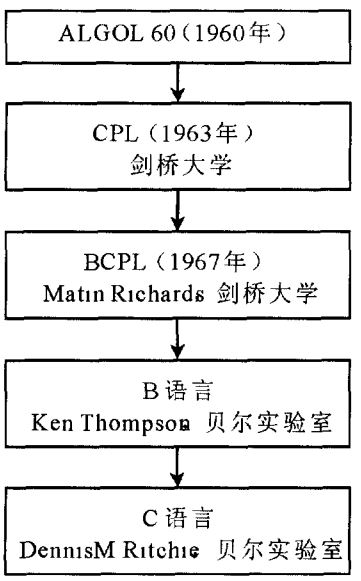

图 1.1　C 语言的发展概况

如图所示，1960 年出现的 ALGOL 60 是一种面向问题的高级语言，与计算机硬件无关，不宜用来编写系统程序。

1963 年英国的剑桥大学推出的 CPL(Combined Programming Language)语言，试图在 ALGOL 60 语言的基础上更接近硬件一些，但由于语言规模比较大，难以学习和实现。

1967 年英国剑桥大学的 Matin Richards 对 CPL 语言作了简化，推出了 BCPL(Basic Combined Programming Language)语言。BCPL 语言保持了 CPL 语言的基本优点。

1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，又作了进一步简化，设计出了非常简单又较依赖硬件的 B 语言(取 BCPL 的第一个字母)，并用 B 语言书写了第一个 UNIX 操作系统，在 DEC PDP-7 机上实现。

但由于 BCPL 语言和 B 语言过于简单，因此语言能力有限，只适用于解决一些特殊类型的问题。从 1971 开始，贝尔实验室的 D. M. Ritchie 在 B 语言的基础上开发了第一个 C 编译程序。因为这个编译程序是在 B 语言的基础上开发的，而且无论是在英文字母序列中，还是在 BCPL 的名字中，排在 B 后面的都是字母 C，所以称之为 C 语言。C 语言既保持了 BCPL 和 B 语言精练、较依赖硬件的优点，又克服了它们过于简单、数据无类型等缺点。

1973 年，K. Thompson 和 D. M. Ritchie 两人合作，把 UNIX 全部用 C 语言又改写了一遍，这为 UNIX 的移植和发展奠定了基础。

之后，C 语言虽多次作了改进，但主要还是在贝尔实验室内部使用。直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们普遍注意。

最初的 C 语言是附属于 UNIX 操作系统的，在小型机 PDP-11 上实现。1977 年出现了不依赖于具体机器的 C 语言编译文本，使 C 语言移植到其他机器的工作大大简化，这同时也推动了 UNIX 操作系统在各种机器上实现。例如，VAX，AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用，C 语言也迅速得到推广。1978 年以后，C 语言已先后移植到大、中、小、微型机上，独立于 UNIX 和 PDP，运行在各种操作系统和机种上。

当时的 C 语言，同现在的标准版本，无论是在功能上，还是在表示方法上，都有所区别。所谓标准版本，一般是指由 Brian W. Kernighan 和 Dennis M. Ritchie 合著的《The C Programming Language》，这本书是基于 1978 年发表的 UNIX 第 7 版中的 C 编译程序而写的。该书中介绍的 C 语言后来成为广泛使用的 C 语言版本的基础，被称为标准 C。

1983 年，美国国家标准化协会(ANSI)根据 C 语言问世以来各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。1987 年，ANSI C 又公布了新标准-87 ANSI C。目前流行的 C 编译系统都是以它为基础的。

1.3　C 语言的特点

与其他高级语言相比，C 语言之所以发展迅速，成为最受欢迎的语言之一，主要原因是它具有强大的功能。归纳起来，C 语言具有下列一些特点：

(1) 语言简洁、紧凑，使用方便、灵活：C 语言只有 32 个关键字，9 种控制语句，程序主要由小写字母组成，书写格式自由。压缩了其他高级语言中的冗余的部分。表 1.1 列出了 C 语言同 PASCAL 语言、BASIC 语言的部分功能比较。

表 1.1　C 语言同 PASCAL 语言和 BASIC 语言的比较

C 语言	Pascal 语言	Basic 语言	含义
{ }	BEGIN? END	无	复合语句
if (条件)　语句	IF(条件)THEN　语句	IF(条件)THEN　语句	条件语句
int x;	VAR X:INTEGER	DIM X	定义一个整型变量
int y[5];	VAR Y: ARRAY[0..4] OF INTEGER	DIM X(5)	定义一个整型数组
int f( )	FUNCTION f(): INTEGER	SUB F	定义返回值为整型的函数
int * p	VAR P:^INTEGER	无	定义一个整型指针
x+=3	x=x+3	x=x+3	赋值语句
i++	i=i+1	i=i+1	自增语句

(2) 运算符丰富，表达式类型多样：C 语言中共有 43 种运算符(见附录 B)，可以实现在其他高级语言中难以实现的功能；表达式类型多样，既提高了编译效率和目标代码的质量，又提高了程序的可读性。

(3) 数据结构丰富：C 语言中的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型等，能够用来实现各种复杂的数据结构(如线性表、链表、栈、队列、树、图等)的运算。

(4) 具有结构化的控制语句：C 语言是一种结构化的程序设计语言，其逻辑结构可以用顺序、选择和循环三种基本结构组成，以函数作为模块，实现程序的模块化设计，符合现代编程风格。结构化语言的特点是代码与数据分隔，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。

(5) 语法限制不太严格，程序设计自由度大：C 语言编译系统语法检查不太严格。例如在 C 语言中对数组下标越界不作检查，由编程者自己保证程序的正确；变量类型使用灵活，整型和字符型变量可以通用等。其优点是允许编程者有较大的自由度。但限制和灵活是一对矛盾，有时编译并没有错误，而结果却是错误的。这就要求编程者在编程时自我约束，养成良好

的、严谨的编程习惯，程序编好后要仔细检查。

(6) C语言是中级语言：C语言允许直接访问计算机的物理地址，能对位进行操作，能够实现汇编语言的大部分功能，即可以直接对硬件进行操作。由于C语言既具有高级语言的基本结构和语句，又具有低级语言的实用性，因此人们称之为“高级语言中的低级语言”或“中级语言”。

(7) 生成的目标代码质量高：C语言生成的目标代码只比汇编语言生成的代码效率低20%左右，故程序执行的效率高。

(8) C程序的可移植好：用C语言编写的源程序基本上不作或稍作修改就可在其他型号的计算机上运行。

1.4　简单的C程序举例

学习一种新的程序设计语言，最好是先通过阅读程序来掌握语言的语法规则。下面从一个非常简单的程序开始予以介绍。

例1.1：

```
main()
{
printf("This is a C program.\n");
}
```

将程序输入计算机，编译、连接后执行，在屏幕上会输出下面一行信息：

This is a C program.

通过阅读这段程序，应掌握以下几点：

(1) C程序由函数构成。最简单的C程序只包括一个函数，函数名为main()，称为主函数。一个C程序可以包括多个函数，但有且只有一个main( )函数。程序总是从main()函数开始执行。

(2) 函数名后的一对括号中可以包含若干个参数，这里main()函数不带任何参数，但是括号仍然保留。

(3) 大括号{ }中的部分称为函数体，用来标识程序的开始和结束。

(4) 主函数main()仅包含一个语句，语句由printf输出函数构成。printf函数是标准I/O函数库提供的函数，用于实现输出操作。语句必须以分号(;)结尾。

例1.2： 求两数之和。

程序为：

```
main()
{
int a,b,sum;                /* 定义三个整型变量 */
a=100; b=50;
sum=a+b;
printf("sum is %d\n", sum);
}
```

将程序输入计算机，编译、连接后执行，程序将求出两整数a、b之和sum，并在屏幕上输出

sum 的值。与例 1.1 相比，程序多了几条语句，下面具体解释一下：

(1) "/*　*/"中间的内容表示"注释"。注释是程序员对程序某部分的功能和作用所做的说明，是给人看的，对编译和运行不起作用。在程序中应尽量使用注释语句，以增强程序的可读性。注释可以加在程序的任何位置。

(2) 第 3 行"int a,b,sum;"定义了三个整型变量 a,b,sum。

C 程序中的语句分定义和执行两大部分，定义部分的作用是为变量分配合适的存储单元。C 程序中的变量都必须遵循先定义后使用的原则。

(3) 第 4 行是两个赋值语句，分别把常数 100 和 50 赋给变量 a 和 b。

C 语言规定，一个语句可以占多行，一行也可以有多个语句。

(4) 第 5 行也是一个赋值语句，语句"sum=a+b;"先计算变量 a、b 之和，并把和值赋给变量 sum。

(5) 第 6 行用 printf 函数输出变量 sum 的值。输出函数包括输出格式和变量两部分。其中，输出格式部分为双引号引出的字符串。字符串由两部分组成，一部分是直接显示的字符串，另一部分是以%开始的字符，它表示对应变量的输出格式. 如%d 表示整型格式，在输出时，该位置用对应变量 sum 的值代替。

例 1.2 程序的输出结果为：sum is 150

例 1.3：从键盘输入两个整数，在屏幕上输出它们的最大值。

程序为：

```
main()
{
int a,b,c;
scanf("%d,%d",&a,&b);
c=max(a,b);
printf("max=%d",c);
}
int max(x,y)                /* 用户自定义函数,求两整数的最大值 */
int x,y;
{
int z;
if (x>y) z=x;
else z=y;
return z;
}
```

将程序输入计算机，编译、连接后执行。程序将求出两整数 a、b 的最大值，并在屏幕上输出。

程序说明：

(1) C 语言是一种结构化的程序设计语言，为了使程序的结构清晰，往往将程序按功能不同划分为若干模块，以函数形式构成 C 程序。这个程序包括两整数 a、b 的数据输入，求 a、b 的最大值，最大值输出等功能。按结构化的程序设计思想，程序包括两个函数：主函数 main()和函数 max。其中：主函数完成数据的输入、输出操作；函数 max 为用户自定义函数，用于求两

整数的最大值.通过主函数调用实现两整数 a、b 最大值的求值。

(2) 一个用户自定义函数由两部分构成：

① 函数的说明部分，包括：函数名、函数类型(返回值类型)、函数属性、形式参数名、形式参数类型。

② 函数体，大括号{ }中的部分。函数体中包含变量定义部分和执行部分。

下面是例 1.3 程序中的自定义函数部分的程序及其说明：

```
函数类型   函数名   函数参数
↓
int   max( x, y )                 }
int   x, y;      ←— 函数参数说明  } 函数说明

{ int   z;        ←— 变量定义     }
  if(x>y) z=x;                    }
  else   z=y;                     } 函数体
  return z;                       }
}
```

(3) 主函数 main()中的 scanf 函数为输入函数。scanf 函数也是标准 I/O 函数库提供的函数，用于实现输入操作。程序中 scanf 函数的作用是输入变量 a，b 的值，&a 和 &b 中的 & 为取地址，表示将输入的数据存入变量所指定的存储单元内。输入函数包括输入格式和变量地址两部分。其中，输入格式部分为双引号(" ")引出的字符串.同样，以%开始的字符表示对应变量的输入格式，如%d 表示整型格式。

(4) 主函数 main()中的第 4 行用来调用 max 用户自定义函数，通过用实际参数 a、b 代替形式参数 x，y 实现求两整数 a、b 最大值操作。具体函数调用过程，以及实际参数和形式参数的概念，将在以后有关章节中介绍。

通过以上三个例子，可以总结出如下几点：

(1) C 程序由函数构成，一个 C 程序可以只包括一个主函数 main()，也可以包括多个函数，主函数名为 main()，但有且只有一个 main()函数。函数是 C 程序的基本单位。C 程序中有两种类型的函数：

① 系统提供的标准函数库的函数，如 printf、scanf 等。Turbo C 开发系统提供了 300 多个库函数，用于实现特定的功能，如输入输出、数学计算等。

② 用户自定义函数，如 max()。

(2) 函数包括函数说明和函数体两大部分，并均由语句构成。语句必须以分号(;)结尾。C 程序书写格式自由，一个语句可以占多行，一行也可以有多个语句。

(3) 不管主函数 main()在源程序中的位置如何，一个 C 程序总是从主函数 main()处开始执行的。

(4) C 语言没有输入输出语句，其输入输出操作是通过调用函数来完成的。

(5) C 语言用程序中"/*　*/"作注释。

1.5　C 程序的上机步骤

1. C 程序的上机步骤

上机运行一个 C 程序，必须经过以下三步：

1）编辑 C 源程序文件

所谓 C 源程序文件，就是存放 C 源程序的文件。C 源程序文件可以随意命名，但扩展名必须是. C(如 A. C)。

编辑 C 源程序文件有两种含义：建立或修改 C 源程序文件。如果磁盘中没有相应的文件，则编辑的目的就是建立此文件，将程序输入到文件中；如果相应的文件已经存在，则编辑的目的就是要修改此文件。

2）编译和连接

(1)编译：计算机不能直接识别源程序。要执行源程序，首先必须将其翻译成机器语言程序，即目标代码程序，计算机才能直接识别。编译是一种翻译方式，编译工作是由计算机运行专门的编译程序自动完成的。编译后得到的目标代码文件的扩展名为. OBJ(如 A. OBJ)。

(2) 连接：如果编译后得到的目标代码中使用了一些未在本目标模块中定义的外部引用，如系统函数，则还必须把有关的各种代码装配在一起产生一个完整的可执行文件后，才能执行。组合和装配各种代码的过程就称为连接，由专门的连接程序完成。连接后得到的文件称为可执行文件，其扩展名为. EXE(如 A. EXE)。

3）执行程序

经过编译和连接，最后得到了扩展名为. EXE 的可执行文件，这就可以直接拿来运行了。

对于以上三个步骤，其每一步骤都有相应的程序命令，但不同版本的 C 语言其命令会有所不同。本书中，选用 Turbo C 2.0 作为上机实习工具。Turbo C 集成环境是一个集编辑、编译、连接、调试、运行和文件管理为一体的工具，对于 C 程序上机过程的三个步骤，都可在此集成环境中完成。下面首先介绍一下 Turbo C 集成环境的使用。

2. Turbo C 集成环境的启动

启动 Turbo C 之前，首先要将系统安装在指定的目录下，然后执行相关的命令将 TC. EXE 加载到内存中。以下是启动 TC 集成环境的几种情况。

1）单机 DOS 环境下启动 TC

如果使用的计算机是单机，并准备在 DOS 环境下启动 TC 集成环境，则首先要弄清 TC 安装的位置，一般 TC 集成环境安装在名为 TC 的目录下，找到此目录。然后，将当前工作目录修改到此目录下，在 DOS 提示符下输入命令 TC，启动 TC 集成环境。

2）Windows 下启动 TC

如果要在 Windows 环境下启动 TC，则可通过“我的电脑”找到 TC 图标，或在“资源管理器”中找到 TC. EXE，用鼠标双击来启动 TC 集成环境。

启动 TC 后，屏幕首先出现 TC 主屏幕和版本信息。按任意键可使版本信息消失，这时系统进入 Turbo C 编辑环境，如图 1.2 所示。其中，最上面一行为 Turbo C 2.0 主菜单，中间窗口为编辑区，接下来是信息窗口，最底下一行为参考行。

此为试读，需要完整PDF请访问：www.ertongbook.com