普通高等教育“十一五”国家级规划教材

郝莹 刘亚姝 孙雷 编著

C语言程序设计教程

21世纪计算机科学与技术实践型教程

丛书主编 陈明

清华大学出版社

普通高等教育“十一五”国家级规划教材

郝莹 刘亚姝 孙雷 编著

C语言程序设计教程

清华大学出版社
北京

21世纪计算机科学与技术实践型教程
丛书主编 陈明

内 容 简 介

本书是从结构化程序设计的角度来编写的，各章节结构紧凑，前后衔接紧密。在内容上，改变传统C程序设计图书中各控制流程占据过多篇幅而不能突出结构化、模块化程序设计的安排，将“控制流程”的语法内容集中在一个章节中，而将各种“控制流程”的应用在随后章节中体现。为了能够强化结构化程序设计方法，本书以较大篇幅介绍“函数”，并在每个实例中分析其功能模块的结构设计。“指针”是C语言功能强大的体现，也是教学中的难点，为了使读者更好地理解“指针”的概念，书中着重分析了指针与“数组”、函数以及字符串的关系。此外，为了设计优良的程序，也将数据结构的概念在“结构体”这一章中体现，并介绍简单数据结构及应用，为不同专业的学生后续学习打下良好的基础。书中各章还附有习题。

本书主要针对高等院校建筑类专业的学生编写，在实例中体现了其专业的应用领域特色。

图书在版编目（CIP）数据

C语言程序设计教程/郝莹，刘亚姝，孙雷编著. —北京：清华大学出版社，2009.10
（21世纪计算机科学与技术实践型教程）
ISBN 978-7-302-20637-8

Ⅰ. C… Ⅱ. ①郝… ②刘… ③孙… 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(2009)第123320号

责任编辑：谢　琛　张为民
责任校对：焦丽丽
责任印制：何　芊

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京季蜂印刷有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185×260　　**印　张**：20.5　　**字　数**：471千字
版　　次：2009年10月第1版　　**印　次**：2009年10月第1次印刷
印　　数：1～5000
定　　价：29.50元

《21世纪计算机科学与技术实践型教程》

编辑委员会

《21世纪计算机科学与技术实践型教程》

序

21世纪影响世界的三大关键技术：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制定科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21世纪计算机科学与技术实践型教程》就是为了填补这部分空白。将理论与实际联系起来，使学生不仅学会了计算机科学理论，而且也学会应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，他们把积累的经验、知识、智慧、素质融合于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编　陈明

2005年1月于北京

前　言

C语言是广泛使用的一种程序设计语言，也是高等学校计算机基础教育中程序设计课程的必选之一。C语言的功能丰富，表达方式灵活，但C语言涉及的概念也相对比较复杂，规则多，使用时容易出错。

作为一本C语言程序设计类教材，与国内传统书籍“针对语言而讲语言”不同，本书提供了丰富的与实际联系紧密的，特别是与建筑行业相关专业应用相结合的实例，使读者更易于掌握这门语言。

本书主要内容包括C语言概述、数据类型及基本输入输出、控制流程、数组、函数、指针、结构体、预编译命令、输入与输出等。

本书主要特点如下：

(1) 基于现代程序设计方法，从结构化程序设计的角度来编写；

(2) 各章节力争做到结构紧凑，前后内容衔接紧密；

(3) 突出结构化、模块化程序设计的特点；

(4) 以较大篇幅介绍“函数”，并将控制流程结合在函数中应用；

(5) 将数组与指针结合起来，有利于理解指针的概念以及其与数组的关系；

(6) 将数据结构的概念在“结构体”这一章中体现；

(7) 为了给学生后续的学习打下良好的基础，在结构体、指针等章节的基础上介绍简单数据结构；

(8) 介绍了有关“预编译命令”和“输入与输出”的相关内容。

此外，学习程序设计的过程中重要的是算法的学习，因此，本书从“控制流程”之后引入算法设计和表示方法，在每一个实例编写之前介绍了设计程序的模块以及程序流程。

由于作者水平有限，书中不足之处在所难免，敬请批评指正。

作　者

2009年4月

目　录

第1章 C语言概述

C语言是国际上广泛流行的计算机高级程序设计语言之一，它既可以用来编写应用软件，也可以用来编写系统软件，而且在嵌入式系统的编程中也有广泛的应用。

1.1 C语言的发展过程

计算机语言的发展是一个不断演化的过程，从最初的机器语言到汇编语言再到各种结构化的高级语言，最后发展到支持面向对象技术的面向对象语言。程序设计语言的发展是不断地把机器能够理解的语言提升到模仿人类思考问题的形式语言。

最早期使用的是二进制语言，程序设计人员只能使用计算机直接识别和执行的二进制代码来编写程序。例如，在Intel 8086 CPU的计算机中，用机器指令完成100和256的加法，需要编写如下代码：

```
10111000  01100100  00000000
00000101  00000000  00000001
```

可以看到，直接用机器指令编写程序是一项非常烦琐的工作。为了减轻程序设计人员的负担，很快出现了汇编语言，这种语言是用符号来代表二进制代码，所以称为符号语言。例如，在上例中，用汇编语言完成100和256的加法，只需要如下代码：

```
MOV  AX, 100
ADD  AX, 256
```

其中ADD用来表示“+”，可以看到用符号代替二进制的机器指令已经大大简化了程序的编写工作，然而用这种语言编写的程序需要通过一种软件翻译后才能执行。

不管是二进制语言还是汇编语言，所编写的程序一般只能在同类型的计算机上运行，所以这种语言又称为“面向机器的语言”。

程序设计的关键是将问题及解决问题的算法过程描述出来。设计人员很快就提供了一种描述算法过程很方便，同时脱离对机型的要求，能在任何计算机上运行的计算机语言。程序设计人员可以利用这种语言直接写出各种表达式来描述简单的计算过程。专家们将这种语言称为“高级语言”，而将二进制语言和汇编语言统称为“低级语言”。由于高级语言是面向问题和算法过程描述的，所以又将高级语言称为“面向问题的语言”。

世界上第一个高级语言是 ALGOL，也叫算法语言。C 语言的前身就是 ALGOL 语言。1960 年，ALGOL 60 版本推出后，很受程序设计人员的欢迎。用 ALGOL 60 来描述算法很方便，但是它离计算机硬件系统很远，不宜用来编写系统程序。1963 年，英国剑桥大学在 ALGOL 语言基础上增添了处理硬件的能力，并命名为 CPL（Combined Programming Language，复合程序设计语言）。CPL 由于规模大，学习和掌握困难，并没有流行起来。1967 年，剑桥大学的马丁·理查德对 CPL 语言进行了简化，推出 BCPL（基本复合程序设计语言）。1970 年，美国贝尔实验室的 Ken Thompson 对 BCPL 进行了进一步的简化，突出了硬件处理能力，并取了 BCLP 的第一个字母 B 作为新语言的名称。同时用 B 语言编写了 UNIX 操作系统程序。1972 年，贝尔实验室的 D. M. Ritchie 对 B 语言进行了完善和扩充，在保留 B 语言强大硬件处理能力的基础上，扩充了数据类型，恢复了通用性，用 C 作为新语言的名称。此后 1973 年，Ken Thompson 与 D. M. Ritchie 合作将 UNIX 操作系统的 90%用 C 语言来编写，即 UNIX 第 5 版。随着 UNIX 6 操作系统的成功发布，C 语言也很快成为一种很受欢迎的计算机语言。

1977 年，为了让 C 语言脱离 UNIX 操作系统，出现了不依赖于具体机器的 C 语言编译文本，使得 C 语言移植到其他机器的工作大大简化。1978 年，C 语言已经可以移植到任何机器上，独立于 UNIX 了。1978 年，Brain W. Kernighan 和 Dennis M. Richie 合著了流行世界的 *The C Programming Language* 对 C 语言的语法进行了规范化的描述，成为当时的标准，即标准 C。随着微型计算机的普及，出现了许多不同的 C 语言版本，为了统一标准，美国标准化协会（ANSI）于 1983 年开始，制定了 C 语言的标准，简称为 ANSI C。ANSI C 与标准 C 相比有了很大发展。1987 年，ANSI 又在 83 ANSI C 的基础上发布了新的标准，即 87 ANSI C。1989 年，ANSI C 被完全采纳并于 1990 年推出第一个范本，国际标准化组织（ISO）也采用此标准，该标准一般被称为 ANSI/ISO Standard C。1988 年，Brain W. Kernighan 和 Dennis M. Richie 按照 ANSI C 标准修订了 *The C Programming Language* 一书。1989 年的 C 标准称为标准C++ 的基础文档，并定义 C 为C++ 的子集。1989 年制定的 C 标准简称为 C89。

进入 20 世纪 90 年代，又新出现了许多高级程序设计语言，但是 C 标准的修订一直在进行，最终形成了 1999 年的 C 标准，简称为 C99。

1.1.1 C 语言的地位

C 语言通常被称为高级语言，但由于它自身的特点也被称为中级语言，因为它比其他高级语言更接近硬件，比低级语言更接近算法，C 语言程序易编、易读、易查、易改。表 1-1 给出了 C 语言在计算机程序设计语言中的地位。

表 1-1 C 语言的地位

高级语言	Pascal、Delphi、VB、Java、C#
中级语言	C++ 、C
低级语言	汇编语言

C 语言允许对位、字节和地址等进行操作，而这些正是低级语言的特性——操作硬件

方便。因此,C 语言很适合编写系统程序,目前很多嵌入式系统都采用 C 语言作为开发语言。

与其他高级语言一样,C 语言中定义了丰富的数据类型和运算符,可以很方便地描述程序设计的算法。常用的数据类型有整型、实型以及字符型,C 语言允许这些类型之间的相互转化,特别是整型与字符型可以混合在一起写在表达式中,类型的转换比较灵活。这一特点也反映在函数调用中,参数的类型不一定要求完全一致,C 语言甚至能够完成相容类型的自动转化。而高级语言一般是要求类型必须相同才能进行参数传递。

与低级语言相比,C 语言能够用非常简单的 32 个保留字编写出所有的程序设计代码,而不需要记住如汇编语言那样的助记符。

由于 C 语言兼具了高级语言与低级语言的特性,因此,在日渐繁多的程序设计语言中永远占有一席之地,而且,通常是作为学习程序设计的入门语言来学习。

1.1.2 C 语言的特点

C 语言发展迅速,而且成为最受欢迎的语言之一,主要是因为它具有不同于(或优于)其他语言的特点。C 语言的主要特点有以下几点:

(1) 功能强大,兼具高级语言与低级语言的特性。C 语言伴随着 UNIX 诞生之初就决定了它在编写系统软件上的优势,有许多著名的系统软件,如 DBASE Ⅲ PLUS、DBASE Ⅳ 都是由 C 语言编写的。而用 C 语言加上一些汇编语言子程序,就更能显示 C 语言的优势了,例如 PC-DOS 、WORDSTAR 等就是用这种方法编写的。

(2) 数据类型和运算符十分丰富,程序设计和算法描述更为简单和方便。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。能用来实现各种复杂的数据类型的运算。特别是引入了指针概念,使程序效率更高,能够实现复杂的数据结构,例如,链表、树等。它有 34 种运算符,能够将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,使得 C 语言可以如汇编语言一样对位、字节和地址进行操作。

(3) 语法结构简洁紧凑、简单易学。C 语言一共只有 32 个关键字、9 种控制语句,简单易学。关键字如下:auto、break、case、char、const、continue、default、do、double、else、enum、extern、float、for、goto、if、int、long、register、return、short、signed、sizeof、static、struct、switch、typedef、unsigned、union、void、volatile、while。控制语句如下:if()…else…、for()…、while()…、do…while()、continue、break、switch、goto、return。

C 语言程序书写自由,语法限制不严格,程序设计自由度大。例如,C 语言对数组的下标越界并不做语法检查,要由编程人员自己保证程序的正确性。程序设计人员需要在学习过程中不断地深入理解 C 语言的标准,更快地掌握 C 语言的精髓。

(4) 具有结构化控制语句,适合结构化程序设计。C 语言提供了如 if…else、while 等语句,能够实现分支、循环等结构化程序的设计。

(5) 是一种模块化程序设计语言。C 语言提供了函数作为程序设计的基本模块单位,便于实现程序的模块化,适合大型软件的研制和调试。

(6) 提供了大量的库函数,简化了程序设计工作。C 语言提供了许多常用库函数,例

如数学类的 math.h 头文件，其中包含了丰富的数学运算函数，使用时只需要简单地调用就可以，大大简化了程序设计工作。

(7) 程序生成代码质量高，执行效率高。C 语言一般只比汇编程序生成的目标代码效率低 10%～20%，执行效率很高。

(8) 适用范围大，可移植性好。1977 年就出现了 C 语言的可移植编译程序，使得 C 语言程序代码的编译可以独立于硬件。目前，C 语言的突出优点就是适合于多种操作系统，如在 Windows 下编写的程序不需要或者只需要简单地改动就能在 UNIX 下运行，同时也不局限于某一机型，而是适用于多种机型。

1.2 编写一个简单 C 语言程序

程序设计的学习通常从案例开始，下面介绍一个简单的“Hello World!”程序是如何编写的。

例 1.1 在屏幕上输出“Hello World!”。

```
1    /*example 1-1.c output a string*/
2    #include <stdio.h>
3    void main(){
4        printf("Hello World!");
5    }
```

运行结果：

```
Hello World!
```

程序第 1 行是注释语句，该语句并不参与运行而是对程序的一个说明；第 2 行是预编译命令能够将待用到的与 I/O 相关的函数包含进来；第 3 行是函数名及“{”表示程序的开始；第 4 行是一条语句，调用 printf 函数输出“Hello World!”信息；最后一行“}”表示程序的结束。

从这个例子中，可以看到一个简单的 C 语言程序由以下几部分构成：

(1) 由函数构成：一个可执行的 C 语言程序必须有一个而且只能有一个称作主函数的 main 函数；程序体必须用一对花括号“{”和“}”括起来，如例 1.1 中的第 3 行及第 5 行；程序的执行由 main 函数的第一行开始到 main 函数结束。

(2) 程序体由语句构成，每个语句的结尾必须要用“;”作为终止符。C 语言程序书写格式自由，多条语句可以写在一行上，C 语言程序没有行号的概念。为了程序的可读性，C 语言程序的书写上要注意缩进，如图 1-1 所示。

(3) C 语言本身没有输入输出语句。需要采用第 2 行中的 #include<stdio.h>语句将输入和输出的库函数包含进来，这种方式就使得 C 语言本身的规模较小，编译程序更简单，具有可移植性。

(4) C 语言程序可以包含注释。注释是包含在“/*”和“*/”之间的内容，在编译时它被 C 语言编译器忽略。一个好的、优秀的程序员要养成添加注释的习惯，必要的注释