21世纪高等院校规划教材

C语言程序设计教程

THE C PROGRAMMING LANGUAGE

陈　波　吉根林　主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21世纪高等院校规划教材

C语言程序设计教程

主编　陈　波　吉根林

编著　王　琼　周俊生　于　泠

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书面向初学者，立足C99标准，重点介绍了与C89兼容的内容。全书共分10章：C语言概述，数据类型，基本语句与结构化程序设计，数组，函数和模块化程序设计，指针，编译预处理，结构体、共用体和枚举类型，文件以及位运算。

本书集作者多年“C语言程序设计”课程的教学经验，全书体系完整，内容由浅入深，条理清晰，语言流畅；每章附有较多的图表，使读者能够准确、直观地理解问题；样例丰富，紧扣知识点，并以Visual C++ 6.0为程序平台，可操作性强；注重编程方法与技巧的讲解，重视对编程能力的培养；每章均附有习题和上机实验题，习题覆盖知识重点，题型丰富。书后附录提供了两套笔试和上机模拟试卷，做到了教材、实验、习题三位一体。

本书适合作为高等学校计算机专业及相关专业C语言程序设计课程的教材，也可作为计算机等级考试参考书，还可供从事计算机软件开发人员参考使用。

图书在版编目（CIP）数据

C语言程序设计教程/陈波，吉根林主编. —北京
：中国铁道出版社，2010.8
21世纪高等院校规划教材
ISBN 978-7-113-11428-2

Ⅰ.①C… Ⅱ.①陈… ②吉… 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（2010）第134410号

书　　名：C语言程序设计教程
作　　者：陈　波　吉根林　主编

策划编辑：秦绪好　杨　勇
责任编辑：秦绪好　　**读者热线电话：400-668-0820**
编辑助理：张　丹
封面设计：付　巍　　**封面制作：李　路**
责任印制：李　佳

出版发行：中国铁道出版社（北京市宣武区右安门西街8号　　邮政编码：100054）
印　　刷：三河市华丰印刷厂
版　　次：2010年8月第1版　　2010年8月第1次印刷
开　　本：787mm×1092mm　1/16　印张：19.75　字数：474千
印　　数：3 000册
书　　号：ISBN 978-7-113-11428-2
定　　价：29.80元

前言

FOREWORD

“C 语言程序设计”是高等学校计算机科学与技术及相近专业中程序设计类最重要的基础课程之一，也是理工科相关专业重要的公共基础课，全国以及各省市的计算机等级考试等都将 C 语言列入了考试范围。学习 C 语言、学好 C 语言是每一个准备从事计算机或相关行业人员的迫切愿望。

本教材的 5 位作者全部是多年从事本科生“C 语言程序设计”课程教学的教师，在多年的 C 语言教学实践中，学生普遍反映难学难懂；听课时懂了，上机时脑子一片空白；笔试时的题目会做，编程题却无从下手；有解答、分析的编程题会了，面对新问题又不知如何解决……

面对诸如此类的问题，几位作者进行了多年的思考和教学实践。本教材就是作者多年从事“C 语言程序设计”课程中，进行教学内容与教学方法改革与探索的结晶。

首先，本书面向初学者，立足 C99 标准，重点介绍了与 C89 兼容的内容，全书体系完整，内容条理清晰，语言流畅。全书共分 10 章：第 1 章介绍程序设计语言的发展、C 语言的发展、特点以及 C 语言的标准，分析了 C 语言源程序的实例结构，给出了上机运行一个 C 程序的步骤；第 2 章介绍了数据类型、常用表达式和运算符；第 3 章介绍了基本语句与结构化程序设计，重点分析了顺序、分支、循环 3 种基本控制结构在程序设计中的应用；第 4 章介绍了数组；第 5 章介绍了函数和模块化程序设计；第 6 章介绍了指针；第 7 章介绍了编译预处理的使用；第 8 章介绍了结构体、共用体和枚举类型；第 9 章介绍了文件的基本操作；第 10 章介绍了位运算操作。

其次，本书内容讲解由浅入深，直观易懂，既重视对知识点的梳理，又重视对编程能力的训练。例如，每章附有较多图表，使读者能够准确、直观地理解问题；样例丰富，紧扣知识点，一些样例在多个章节中用不同的编程方法实现，便于读者对于相关知识在编程中的应用进行比较；每个例题均在 Visual C++ 6.0 开发环境中运行通过，并给出运行结果图，可操作性强；样例中给出编程方法分析以及技巧的讲解；注重对读者良好编程风格的引导，每个例程均增加空行和适当的注释；每章后面还给出了初学者在编程中易犯的错误，使读者在学习中少走弯路。

最后，本书每章均附有习题和上机实验题，习题覆盖知识重点、题型丰富。书后附录提供了两套笔试和上机模拟试卷，做到了教材、实验、习题三位一体。初学者可以通过从书本知识的学习到课后练习再到上机实践，一个层次一个层次地复习、巩固、实践所学内容。

本教材第 1 章和第 10 章由吉根林教授编写；第 2 章由王琼副教授编写；第 3 章和第 7 章由陈波副教授编写；第 4 章和第 6 章由周俊生副教授编写；第 5 章由王琼和陈波共

同编写；第 8 章和第 9 章由于泠副教授编写；附录由陈波和王琼编写。全书由陈波和吉根林担任主编，并负责最后统稿、修改和定稿。本书出版过程中得到了南京师范大学教改项目“计算机专业本科生创新性训练体系建设研究”的支持，以及中国铁道出版社的大力支持，在此表示衷心的感谢！

为了便于教师使用本书教学，本书为任课教师提供电子课件，其中包括全部例题和习题参考解答。

由于编者水平有限，书中难免存在不妥和疏漏之处，敬请读者批评指正。

编者 E—mail：njnu_c@163.com

编 者

2010 年 6 月

CONTENTS

目 录

第 1 章 C语言概述

本章主要介绍程序设计语言的发展；C 语言的发展、特点以及 C 语言的标准，并由浅入深地介绍几个 C 语言源程序的实例结构，使读者可以比较清楚地了解 C 语言程序基本结构与书写规则。最后还给出了上机运行一个 C 语言程序的基本步骤，包括编辑、编译、连接和运行等。

1.1 程序与程序设计语言

1.1.1 计算机与程序

计算机是当今信息化社会中必不可少的工具。它是一种按照事先编写的程序，自动对数据进行输入、处理、输出和存储的系统。计算机要完成不同的工作，就要运行不同的程序。程序就是为完成某项任务而编写的一组计算机指令序列。编写程序的过程称为程序设计，是软件开发的关键步骤，软件的质量主要是通过程序运行的质量来体现的。在进行程序设计之前必须根据实际需求确定使用什么程序设计语言来编写程序。

1.1.2 程序设计语言

人与人之间交流沟通需要使用相互理解的语言，人与计算机交流也要使用相互理解的语言。程序设计语言就是用来实现人与计算机之间交流的语言，它经历了从机器语言、汇编语言到高级语言的发展历程。

1. 第一代语言——机器语言

机器语言是由 0 和 1 组成的指令序列。例如，指令 1011011000000000 表示要计算机执行一次加法操作；而指令 1011010100000000 则表示要计算机执行一次减法操作。它们的前 8 位表示操作码，后 8 位表示地址码。

机器代码可以直接被计算机所识别，显然机器语言最大的特点是效率高、执行速度快。但是采用机器代码编写程序，要求程序员熟记所用计算机的全部指令代码和代码的含义，编写程序时程序员必须自己处理每条指令和每一个数据的存储分配、输入/输出，还要记住编程过程中每步所使用的工作单元处在何种状态。可想而知，用机器语言编写程序是一件十分烦琐且容易出错的工作。

2. 第二代语言——汇编语言

由于用机器语言编程存在工作量大、易于出错等问题，因此人们考虑采用一些简洁的英文

字母、符号串替代特定指令的二进制串，使表达方式更接近自然语言。例如，用“ADD”代表加法，“MOV”代表数据传送等，这样人们就很容易读懂并理解程序在干什么，纠错及维护都很方便。这种采用英文缩写的助记符标识的语言称为汇编语言。

但是，计算机并不认识这些符号，因此需要一个专门的程序，负责将这些符号翻译成二进制数形式的机器语言才能被计算机执行，这种翻译程序称为汇编程序。

汇编语言是一种与机器语言一一对应的程序设计语言，虽然不是用0、1代码编写，但实质是相同的，都是直接对硬件进行操作，只不过指令采用助记符标识，更容易识别和记忆。机器语言和汇编语言均与特定的计算机硬件有关，程序的可移植性差，属于低级语言。由于汇编语言源程序的每一句指令只能对应实际操作过程中一个很细微的动作，如移动、加法等，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，所以人们只有在直接编写面向硬件的驱动程序时才采用它。

3．第三代语言——高级语言

到了20世纪50年代中期，人们研制了高级语言。高级语言是用接近自然语言表达各种意义的“词”和常用的“数学公式”形式，按照一定的“语法规则”编写程序的语言。这里的“高级”指这种语言与自然语言和数学公式相当接近，而且不依赖于计算机的型号，通用性好。高级语言的使用，改善了程序的可读性、可维护性和可移植性，大大提高了编写程序的效率。

用高级语言编写的程序称为高级语言源程序，不能被计算机直接识别和执行，一般都要用翻译的方法把高级语言源程序翻译成目标程序才能执行，这种翻译程序称为编译程序。

高级语言的出现大大简化了程序设计，缩短了软件开发周期，显示出强大的生命力。此后，编制程序已不再只是软件专业人员才能做的事，一般工程技术人员用较短的学习时间，也可以使用计算机解决问题。随着计算机应用日益广泛地渗透到各学科和技术领域，后来发展了一系列不同风格、为不同对象服务的程序设计语言，其中较为著名的有Fortran、BASIC、COBOL、ALGOL、LISP、LP/1、Pascal、C等十几种语言。

1.1.3 高级语言程序的开发过程

有人认为程序设计是一门艺术，而艺术在很大程度上是基于人的灵感和天赋的，往往没有具体的规则和步骤可循。对于一些小型程序的设计，上述说法可能有一些道理。但是，对于大型复杂程序的开发，仅有灵感和天赋是不能很好地解决问题的，人们在几十年的程序开发实践中已证明了这一点。事实上，程序设计是一门科学，程序的开发过程是有规律和步骤可循的。通常，高级语言程序的开发遵循以下步骤。

1．明确问题

用计算机解决实际问题，首先要明确解决什么问题，即做什么。如果对问题没有搞清楚或理解错了，就试图解决它，其结果是可想而知的。

2．算法设计

明确问题之后，就要考虑如何解决它，即如何做。计算机解决问题的方式就是对数据进行处理，因此首先要对问题进行抽象，抽取出能够反映问题本质特征的数据并对其进行描述，然后设计计算机对这些数据进行处理的操作步骤，即设计算法。

3. 选择某种语言进行编程

算法设计完成后，就必须用某种程序设计语言表达出来，即编程实现。现在的程序设计语言很多，用哪一种语言来编程呢？从理论上讲，虽然各种语言之间存在着或多或少的差别，但它们大多数都是基于冯·诺依曼体系结构的，它们在表达能力方面是等价的，因此对于同一个设计方案，可用多种语言实现。

在实际中，决定采用哪一种语言来编程可以考虑以下因素：

- 设计方案。例如，对于采用功能分解的设计方案，用过程式程序设计语言进行编程比较合适；对于面向对象的设计方案，采用面向对象的程序设计语言来实现就比较自然和方便。
- 编程语言效率的高低、使用的难易程度、数据处理能力的强弱等。
- 一些非编程技术的因素。例如，编程人员的个人喜好。

选定编程语言之后，下面就是使用该语言编写程序。对于同一个设计方案，不同的人会写出不同风格的程序。程序设计风格的好坏会影响到程序的正确性和易维护性。程序设计风格取决于编程人员对程序设计的基本思想、技术以及语言的掌握程度。

4. 测试与调试

程序写好之后，其中可能含有错误。程序错误通常有以下3种。

- 语法错误：指程序没有按照语言的语法规则来书写，这类错误可由编译程序来发现。
- 逻辑（或语义）错误：指程序没有完成预期的功能。
- 运行异常错误：指对程序运行环境的非正常情况考虑不周而导致的程序异常终止。

这些错误可能是编程阶段导致的，也有可能是设计阶段甚至是问题定义阶段的缺陷。

程序的逻辑错误和运行异常错误一般可以通过测试来发现。测试方法有很多，例如：

- 静态测试：即不运行程序，而是通过对程序的静态分析，找出逻辑错误。
- 动态测试：即利用一些测试数据，通过运行程序，观察程序的运行结果是否与预期的结果相符。

值得注意的是，不管采用何种测试方法，都只能发现程序的错误，而不能证明程序正确。例如，想要用动态测试技术来证明程序没有错误，就必须输入所有可能的数据来运行程序并观察运行结果，这往往是不可能的，并且也没有必要。测试的目的就是要尽可能多地发现程序中的错误。

测试工作不一定要等到程序全部编写完成后才开始进行，可以采取编写一部分、测试一部分的方式，最后再对整个程序进行整体测试。即先进行单元测试，再进行集成测试。

如果通过测试发现程序有错误，那么就需要找到程序中出现错误的位置和原因，即错误定位。给错误定位的过程称为调试（debug）。调试一般需要运行程序，通过分段观察程序的阶段性结果来找出错误的位置和原因。

5. 运行与维护

程序通过测试后就可交付使用了。由于使用所有的测试方法只能发现程序中的错误，而不能证明程序没有错误，因此，在程序的使用过程中可能会不断发现程序中的错误。在使用过程中发现并改正错误的过程称为程序的维护。程序维护可分成以下3类：

- 正确性维护：指改正程序中的错误。
- 完善性维护：指根据用户的要求使得程序功能更加完善。
- 适应性维护：指把程序移植到不同的计算机平台或环境中。

1.2 C语言的发展和特点

1.2.1 C语言的发展历史

1969年，美国贝尔实验室的Ken Thompson和Dennis Ritchie开始研制UNIX操作系统。UNIX的早期版本是用汇编语言编写的。但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，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。因此，他们决定开发一种高级语言来编写UNIX操作系统。

1970年，Ken Thompson 以BCPL语言为基础，设计出了很简单又能访问硬件的B语言（BCPL的第一个字母）并用B语言对用汇编语言编写的UNIX操作系统进行了部分改写，此时的B语言过于简单，功能有限。1972—1973年，Dennis Ritchie在B语言的基础上设计出了C语言（BCPL的第二个字母）。C语言既保持了BCPL和B语言精练及访问硬件的优点，又克服了它们过于简单和无数据类型等缺点。1973年，Ken Thompson和Dennis Ritchie将UNIX操作系统的90%程序用C语言改写。

多年来，UNIX V系统配备的C语言一直是公认的C语言标准，Brian Kernighan和Dennis Ritchie合著的*The C Programming Language*（Prentice-hall于1987年出版，国内在2004年出版了该书的中文译本《C程序设计语言（第2版）》）书中对其进行了介绍。

在C语言的发展过程中还有两个重要的标准：C89和C99标准。1983年，美国国家标准协会（ANSI）着手制定C语言的标准，于1989年正式被批准为ANSX3.159-1989，一年以后，该标准也被ISO（国际标准化组织）接收，定为ISO/IEC 9899:1990。通常仍称C89标准。

随着C语言的继续发展，1999年C99标准应运而生，C99保持了几乎所有C89的特征。总的来说，C99与C89之间有以下3种变化：

- 在C89基础上增加的特性。其中最主要的是C99增加了5个新的关键字（C89具有32个关键字，而C99达到37个关键字）：_Bool、_Imaginary、restrict、_Complex和inline。增加的其他特性主要包括：变长数组；单行注释；long long int数据类型；可以在语句可能出现的任何地方定义变量；对预处理程序的增加；for语句内的变量声明；柔性数组结构成员；新的库和头文件等。
- 删除了C89中的某些特性。例如，“隐含的int”。在C89中，大多数情况下，当没有明确指定类型标识符时，通常认为其为int类型，但这一点在C99中是不允许的。
  此外C99中还删除了函数的隐含声明。在C89中，如果一个函数在被使用前未被声明，将被视为隐含的函数声明，但C99不支持这一点。
- 修改了C89中的某些特性。例如，放宽的转换限制、扩展的整数类型、增强的整数类型提升规则、对return语句的约束等。

1.2.2　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 语言具有以下一些特点：

1．C 语言是中级语言

C 语言通常被称为中级语言。因为 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，所以人们称为“高级语言中的低级语言”或“中级语言”。例如，C 语言允许直接访问物理地址，可以像汇编语言一样对位、字节和地址进行操作。

2．C 语言是结构化程序设计语言

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。作为一种结构化程序设计语言，其逻辑结构由顺序、选择（分支）和循环 3 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。

3．语言简洁、紧凑，使用方便、灵活

C89 标准定义的 C 语言只有 32 个关键字、9 种控制语句。和 IBM-PC 的 BASIC 相比，BASIC 包含的关键字多达 159 个。C 程序主要由小写字母组成，书写格式自由。C 程序比较简练，源程序短，输入程序时工作量少。

4．运算符和数据结构丰富，表达式多样

C 语言共有 34 种运算符。在 C 语言中把括号、赋值、强制类型转换等都作为运算符处理，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。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。

C 语言提供了各种各样的数据类型，如整型、实型、字符型、数组类型、指针类型、结构体类型等，能够实现各种数据结构，如线性表、链表、栈、队列、树、图等。尤其是指针类型数据，使用起来灵活、多样，程序运行效率更高。

5．语法限制不太严格，程序设计自由度大

C 语言编译系统的语法检查不太严格。例如，在 C 语言中对数组下标越界不进行检查，由编程者自己保证程序的正确；变量类型使用灵活，整型和字符型变量都可以通用等。其优点是允许编程者有较大的自由度。但明显的缺点是增加了程序的不安全因素。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．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，程序执行的效率高。

7．C 语言程序的可移植性好

与汇编语言相比，C 语言程序基本上不做或稍做修改就可在其他类型的计算机上运行。

总之，由于 C 语言的上述特点，使 C 语言越来越受到程序设计人员的重视，在很多领域得到了广泛应用。

当然 C 语言程序也存在一些不足，如运算优先级太多，不便于记忆；类型检验太弱；虽然

转换比较方便，但同时也增加了程序的不安全因素等。*C Traps and Pitfalls*（2002年和2008年该书有中文译本《C陷阱与缺陷》）一书详细介绍了C语言中易导致程序员犯错误的特性，读者不妨一读。本书也在每一章的最后均增加“常见错误分析”，对读者在初学C语言时易犯的错误进行提醒。

1.2.3 C和C++

C++是以C语言为基础的面向对象程序设计语言。对于C89而言，因为C++标准的制定包容了C89的全部内容，所以C++实现了其全部特性。而对于C99而言，C99的部分特性C++并未包含。因此，对C99而言，C++不是C语言的超集，C语言也不是C++的子集。

一般情况下，绝大部分C++编译器可以编译C和C++程序。因此，很多编程者喜欢用C++编译器来编译C程序。这是可行的，但是对于初学C语言的读者来说，一定要注意：C和C++编译器是两种不同的环境，很多特性并不具有通用性。部分C++编译器建立在C89的基础之上，许多C99的新特性未必能够编译通过（当然，也有部分编译器几乎加入了C99的全部新特性）。本教材使用Visual C++ 6.0提供的编译器，只部分支持C99标准，因此本书仍主要以C89标准来编写程序。

还有一点需要注意的是，C语言源程序的扩展名是.c，而C++源程序的扩展名是.cpp。因此，C语言学习者在使用C++工具编译C程序时要注意将源文件的扩展名命名为.c，而不是.cpp。如果扩展名为.cpp，那么编译器将按照C++的要求编译源文件。

1.3 C语言程序的结构

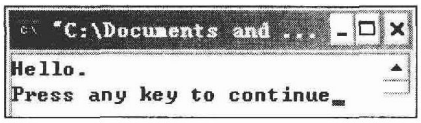
了解和掌握程序的结构是编写程序的基础。一般来说，对于刚开始学习C语言的读者来讲，一个程序可以看做是由函数构成的。为了对程序中的有关内容进行说明，在程序的开头常包含有一些声明。下面通过几个简单的例程初步认识C语言程序的结构。

【例1.1】一个简单的C语言程序。

程序如下：

```
/*This is the first C program.*/
#include<stdio.h>
int main()
{
    printf("Hello.\n");
    return 0;
}
```

在Visual C++集成开发环境中输入程序，经过编译、连接后运行，运行结果截图如下：



说明：

（1）注释。程序的第1行是注释语句。在C语言中，注释是程序员为了增加程序的可读性而增加的说明性信息，对程序的运行不起作用，对源程序进行编译时注释会被忽略。