21世纪高校计算机应用技术系列规划精品教材

丛书主编　谭浩强

C语言程序设计

（第三版）

C Yuyan Chengxu Sheji

(Disanban)

林小茶　编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

21世纪高校计算机应用技术系列规划精品教材

丛书主编 谭浩强

C语言程序设计

（第三版）

林小茶 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

C 语言是程序员的入门语言，也是许多大学为学生安排的第一门程序设计课程。本书充分考虑到这一点，在内容的编排上尽量符合初学者的要求，在实例的选择上从易到难，循序渐进，并且能够解决一些实际问题。

本书的主要内容包括：C 语言概述、C 语言的基本知识、顺序和选择结构程序设计、循环结构程序设计、函数、数组、指针、结构体等构造数据类型和文件。全书通过大量的实例讲解了用 C 语言进行结构化程序设计的要领。

为了方便读者参加全国计算机等级考试二级 C 语言程序设计的考试，本版教材将最新的一次考试的题目分门别类，按照对应的学习内容在第 2～8 章的最后一节做了讲解。

本书适合作为高等学校应用型本科学生学习 C 程序设计课程的教材，也可作为高职高专学生学习 C 语言程序设计课程的教材以及 C 语言自学者的教材或参考书。

图书在版编目（CIP）数据

C 语言程序设计/林小茶编著. --3 版. --北京：
中国铁道出版社，2010.12
21 世纪高校计算机应用技术系列规划精品教材
ISBN 978-7-113-11914-0

Ⅰ. ①C… Ⅱ. ①林… Ⅲ. ①C 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 179254 号

书　　名：C 语言程序设计（第三版）
作　　者：林小茶　编著

策划编辑：崔晓静
责任编辑：杜　鹃　徐盼欣　　**读者热线电话：**400-668-0820
封面设计：付　巍　　**封面制作：**白　雪
版式设计：于　洋　　**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：北京海淀五色花印刷厂
版　　次：2004 年 6 月第 1 版　2007 年 7 月第 2 版　2010 年 12 月第 3 版　2010 年 12 月第 12 次印刷
开　　本：787mm×1092mm　1/16　**印张：**19.75　**字数：**470 千
印　　数：5 000 册
书　　号：ISBN 978-7-113-11914-0
定　　价：30.00 元

21世纪高校计算机应用技术系列规划精品教材

编委会

序

PREFACE

21世纪是信息技术高度发展且得到广泛应用的时代，信息技术从多方面改变着人类的生活、工作和思维方式。每一个人都应当学习信息技术，应用信息技术。人们平常所说的计算机教育其内涵实际上已经发展为信息技术教育，内容主要包括计算机和网络的基本知识及应用。

对多数人来说，学习计算机的目的是为了利用这个现代化工具处理面临的各种问题，使自己能够跟上时代前进的步伐，同时在学习的过程中努力培养自己的信息素养，使自己具有信息时代所要求的科学素质，站在信息技术发展和应用的前列，推动我国信息技术的发展。

学习计算机课程有两种不同的方法：一是从理论入手；二是从实际应用入手。不同的人有不同的学习内容和学习方法。大学生中的多数人将来是各行各业中的计算机应用人才。对他们来说，不仅需要“知道什么”，更重要的是“会做什么”。因此，在学习过程中要以应用为目的，注重培养应用能力，大力加强实践环节，激励创新意识。

由于全国各地区、各高等院校的情况不同，需要有不同特点的教材以满足不同学校、不同专业教学的需要。因此，在教材建设上应当提倡百花齐放，推陈出新。社会应当提供不同内容、不同风格的教材，供各校选用。

根据培养应用型人才的需要，我们组织编写了这套“21世纪高校计算机应用技术系列规划精品教材”。顾名思义，这套教材的特点是突出应用技术，面向实际应用，强调培养应用能力，学以致用。在选材上，根据实际应用的需要决定内容的取舍，重视实践环节，不涉及过多的理论，坚决舍弃那些现在用不到、将来也用不到的内容。在叙述方法上，采取“提出问题—解决问题—归纳分析”的三部曲，这种从实际到理论、从具体到抽象、从个别到一般的方法，符合人们的认知规律，且在实践过程中已取得了很好的效果。本丛书可以作为应用型大学的计算机应用技术课程的教材以及高职高专学校的计算机教材，也可作为广大计算机爱好者学习计算机知识与应用的自学教材。

本丛书采取模块化的结构，根据不同的需要分别组织教材，提供了一个课程菜单供各校选用。以后根据信息技术的发展和教学的需要，不断地补充和调整。我们的指导思想是面向应用、面向实际、面向对象。只有这样，才能灵活地满足各类学校发展的需要。希望各校的老师把你们的要求反映给我们，我们将会尽最大的努力满足。

本丛书自2003年出版以来，受到了广大高校师生的欢迎，至2009年底，已出版了70多种，发行了200多万册，其中《计算机应用基础》一书出版7年内发行了66万册。丛书中有多种教材被国家教育部评为**普通高等教育“十一五”国家级规划教材**。这表示了读者和社会对本丛书的充分肯定，对我们是有力的鞭策。

本套教材由浩强创作室与中国铁道出版社共同策划，选择有丰富教学经验的普通高校老师和高职高专院校的老师编写。中国铁道出版社以很高的热情和效率组织了这套教材的出版工作。在组织编写及出版的过程中，得到全国高等院校计算机基础教育研究会和各高等院校老师的热情鼓励和支持，对此谨表衷心的感谢。

本丛书如有不足之处，请各位专家、老师和广大读者不吝指正。希望通过本丛书的出版，能为我国计算机教育事业的发展和人才培养做出贡献。

全国高等院校计算机基础教育研究会荣誉会长
“21世纪高校计算机应用技术系列规划精品教材”丛书主编

谭浩强

第三版前言

FOREWORD

本教材的第一版和第二版已经印刷了多次，根据读者的需求，本书进行了第三次修订。

本书延续了前两版的特点，对所讲解的内容做了一些调整，以适合第一次学习程序设计人员的需求。

第一，在内容的选择上更适合初学者的需求。例如，链表是 C 语言程序设计的难点，许多学生都反映链表的程序不能理解，因此，本书只讲述单链表的基本概念和操作要点，让学生对链表有一个基础的了解，而比较深入的内容留到“数据结构”这门课程；又如，位运算是 C 语言程序设计的另一个难点，但是如果学过“汇编语言”，那么这部分内容是很容易掌握的；再如，用文本方式显示数据的目的只是为了给调试程序带来方便，花大量的精力学习如何显示格式是没有必要的。

第二，强调好的程序设计思想，不能拘泥于小的程序设计技巧，而破坏了整个程序的清晰度。例如，本书会告诉学生避免使用像“i+++++i”这样的表达式，而不会花大量的篇幅去分析这个表达式到底等于多少，因为这是毫无意义的。资深程序员建议，使用++运算符的最好方法是只使用单句“i++;”，绝不能将++符号使用在一个复杂的表达式里面。类似地，本书在介绍运算符的优先级和结合性时，也提出使用括号表示运算的先后是最好的方法。

第三，编写一些比较实用的程序。例如，本书中有一个例子随着学习内容的深入，不断地被扩充功能，在选择结构程序设计中只有几行语句，而到文件处理时，这个例子已经被扩充到了100行以上。这些程序几乎用到了我们学过的所有基本数据类型和构造数据类型，编者希望读者通过对这些程序的跟踪分析，提高程序设计的能力。

第四，抛弃那些建立好的与程序设计思想关系不大的内容，选择了更具特色和实用性的程序作为实例。例如，奥运奖牌榜、按个人所得税纳税方式计算纳税额等。

第五，沿用了第二版中好的写作思想。在提出问题的同时给出示例程序，在示例程序中尽量将主要的知识点演示出来，使读者对解决同类问题的程序设计思想有比较全面的认识。在示例程序之后，才是对具体问题的讲解和讨论。

第六，对前两版的写作内容进行了整合。在前两版的教材中，按照惯例将 C 语言的基础知识分为两章讲解，但是在实际应用中发现，如果开始学习的简单知识偏多，学生会感觉有些无聊，因为讲解这些知识的案例几乎没有实用价值，主要是在讨论 C 语言的语法，而不是解决问题的程序设计思想。因此，在本版教材中将必需的基础知识安排在第 2 章讲解，而将其他的基础知识与后续内容相结合，力争实例更合理。这是一次尝试，希望得到大家的认可，也希望大家提出宝贵意见。

第七，在第 2～8 章增加了“典型错误及典型例题”。典型错误汇集了作者在教学过程中发现的学生们最容易犯的错误，并讲解了正确的方法，典型例题则是取自最近的一次全国计算机等级考试二级 C 语言程序设计的试卷，通过分析考点给出正确答案，希望为读者参加该考试奠定良好的基础。

本书能够多次再版，要感谢所有使用本书作为教材和学习参考书的教师和读者，同时再次殷切希望您对本书的内容和编写方法提出宝贵的意见和建议。

由于编者水平有限，疏漏之处在所难免，请广大读者批评指正。

编 者

2010 年 9 月

第二版前言

FOREWORD

本教材由于定位比较准确，因而受到使用者的欢迎，已经重印了五次。作者总结了近几年的教学经验，并听取了专家和读者的意见，对本书作了进一步的修订。

这次的修订不仅在内容上作了一些必要的更新，在写作思想上也作了一些改进，希望本书更能满足初学者的需要。

这种新的写作思想是：在提出问题的同时给出实例程序，并在实例程序中尽量将主要的知识点演示出来，使读者对解决同类问题的程序设计思想有比较全面的认识。在实例程序之后，才给出对具体问题的讲解和讨论。

同时，本书在以下几个方面对第一版的内容做了修订。

（1）增加了一些能够提高程序设计能力的内容，删减了与建立程序设计思想关系不大的内容。例如，增加了“控制循环的实用方法”一节，而删减了有关“赋值表达式的类型转换”的内容。

（2）增加了对程序实例的分析和讲解。本书的特点之一是实用程序较多，但是由于课时的限制，教师在课上不能将所有程序都讲解得非常全面，而初学者自己阅读又有一定的困难，因此增加了这部分的内容。

（3）更换了一些程序实例，选择了更具特色和实用性的程序实例。例如奥运奖牌榜、按个人所得税计税方法计算纳税额等。

（4）增加了一部分习题（题序号带有“*”符号的），并且这部分习题将不会在配套的《C 语言程序设计习题解答与上机指导（第二版）》中给出答案，希望学生能发挥自己的主观能动性进行程序设计。

（5）本书的全部程序都重新在 Visual C++ 6.0 环境下进行了调试，与调试环境有关的内容也相应地调整为 Visual C++ 6.0 环境下的，以适应新的教学要求。并且，除了中文显示问题之外，所有程序仍然能在 Turbo C++ 3.0 环境下运行，以方便习惯使用 Turbo C++ 3.0 软件的师生。

作为本书的姐妹篇，我们将同时出版本教程的《C 语言程序设计习题解答与上机指导（第二版）》，给出本教材习题的参考答案，供读者学习时借鉴和参考。

最后，借此再版机会，向使用本书作为教材和教学参考书的教师和其他所有读者表示衷心的感谢，并殷切希望广大读者对本书的内容和编写方法提出宝贵的意见和建议。

由于编者水平有限，疏漏和不足之处在所难免，请广大读者批评指正。

编　者

2007 年 5 月

第一版前言

FOREWORD

对于从未接触过程序设计语言的人来说，以 C 语言作为入门语言也许是有一定难度的，因为 C 语言的语法比较灵活，有些在其他高级语言（例如 Pascal）中被认定具有编译错误的语句，可以轻而易举地通过 C 编译，但是程序运行的结果却不能达到程序设计者的要求，使学习者觉得 C 语言很难。笔者有多年 C 语言教学的经验，对此感受颇深，大约十年以前，在大学教学中，C 语言一般作为第二门甚至第三门程序设计语言课程，因此学生学习起来是很轻松的，如果教师不能讲一些有深度的内容，反而会使学生觉得教师水平太低。所以，C 语言作为入门语言，其侧重点应有所不同。

首先，在内容的选择上不用做到面面俱到。例如，链表的操作不用讲太多，只要讲清楚单链表的基本概念和操作要点就行了，更多的内容应该留在“数据结构”这门课程中介绍；又例如，有关位操作的程序设计，在“汇编语言”课程之前就是讲了，学生也很难搞清楚，但如果学过“汇编语言”，他们会很容易学会这部分内容；再例如，用文本方式显示数据的目的是为了调试程序，花大量的精力学习显示格式是没有必要的。因此，本书在编排上，作了一些小的改革。

第二，要强调好的程序设计思想，不要拘泥于小的技巧，而破坏了整个程序的清晰度。例如，本书将告诉学生避免使用像“i+++++i”这样的表达式，而不是花大量的篇幅去分析这个表达式到底等于多少，因为这是毫无意义的，资深程序员建议，使用++符号的最好方法是只使用单句“i++;”，绝不能将++符号使用在一个复杂的表达式里面。类似地，本书在介绍运算符的优先级和结合性时也提出使用括号表示运算的先后是最好的方法。

第三，编写一些比较实用的程序。本书中有一个例子随着本书内容的深入，不断地被扩充功能，在选择结构程序设计中只有几行语句，而到文件处理时，这个例子已经被扩充到了 100 行以上。这些程序几乎用到了所有我们学过的基本数据类型和构造数据类型，笔者希望读者能够通过对这些程序的跟踪分析，提高程序设计的能力。

在本书的编写过程中，我们尽量使内容通俗易懂、适于自学、由浅入深、便于理解。

作为本书的姐妹篇，我们将同时出版本教程的习题解答和实验指导一书，给出本教材中所有习题的参考答案，供读者学习时借鉴和参考。

在本书编写和出版的过程中，全国计算机基础教育研究会会长谭浩强教授给予了指导和把关，在此表示最衷心的感谢。

在本书的编写和出版过程中还得到了陈维兴教授的帮助和支持，在此表示诚挚的感谢。

由于编者水平有限，书中不足之处在所难免，请广大读者批评指正。

编　者

2004 年 4 月

目录

CONTENTS

第 1 章 C 语言概述

C 语言是一种通用的程序设计语言，它具有丰富的运算符和表达式，以及先进的控制结构和数据结构。C 语言具有表达能力强、编译目标文件质量高、语言简单灵活、容易移植、容易实现等优点。

1.1 程序与程序设计语言

1.1.1 程序

随着计算机走入寻常百姓家，“程序”已经不再是计算机科学使用的专用词汇了。在日常生活中，我们其实在不断地编写程序并执行，只不过人们并没有明确地意识到而已。举个例子，如果要用全自动洗衣机洗衣服，应该怎么做呢？尽管简单，我们还是按照一般人的习惯来描述一下。

第一步，把脏衣服放进洗衣机；

第二步，打开上水的水龙头并安装好电源插头；

第三步，放入洗衣粉；

第四步，按下洗衣机的开始按钮；

第五步，等待洗衣机洗完衣服（当然，不妨去干点别的什么事情）。在洗衣机提示洗完的蜂鸣声响了以后，就可以从洗衣机中拿出干净衣服去晾晒。

上面所描述的五个步骤，就是人们洗衣服的“程序”。也许不同的人使用的步骤并不完全一样，例如将第一步和第二步互换一下，也同样能将衣服洗干净，所以干一件事的“程序”可以不唯一，这也是计算机程序的一个特点。

对于计算机来说，程序就是由计算机指令构成的序列。计算机按照程序中的指令逐条执行，就可以完成相应的操作。更准确一点，计算机执行由指令构成的程序，对提供的数据进行操作。计算机程序的操作对象是“数据”。这里的数据不是简单的阿拉伯数字，而是包括了各种现代计算机能够处理的字符、数字、声音、图像等。

实际上计算机自己不会做任何工作，它所做的工作都是由人们事先编好的程序来控制的。程序需要人来编写，使用的工具就是程序设计语言。

1.1.2 程序设计语言

目前，通用的计算机还不能识别自然语言，只能识别特定的计算机语言。

计算机语言一般分为低级语言和高级语言。

低级语言直接依赖计算机硬件，不同的机型所使用的低级语言是完全不一样的。高级语言则不再依赖计算机硬件，用高级语言编写的程序可以方便地、几乎不加修改地用在不同类型的计算机上。

需要强调的是，无论采用何种语言来编写程序，程序在计算机上的执行都是由 CPU 所提供的机器指令来完成的。机器指令是用二进制表示的指令集。每种类型的 CPU 都有与之对应的指令集。

1. 低级语言

低级语言包括机器语言和汇编语言。

直接使用二进制表示的指令来编程的语言就是机器语言。使用机器语言编写程序时，必须准确无误地牢记每一条指令的二进制编码。如果程序员面对的是“101110001110100000000011”这样的编码序列，能不头痛吗？而且，有时还要求把这些二进制编码再转换成八进制或十六进制数才能输入计算机。这不但加大了程序员的工作量，而且还增加了程序出错的机会，将大量的二进制编码序列准确地转换成八进制或十六进制数，可不是一件容易的事。

机器语言的优点是执行速度快，并且可以直接对硬件进行操作，例如主板上的 BIOS 及一些设备的驱动程序等。

机器语言的缺点也是显而易见的。首先是可读性差，即便是编写程序语句的人，也未必马上就能看懂“101110001110100000000011”表示的是什么命令；其次是可维护性差，其他程序员编写的程序（甚至是程序员自己编写的）很难看懂，如何谈维护呢？再者，就是可移植性差，因为不同的机型有自己的一套机器指令，与其他机型的机器指令不兼容。另外，用机器语言编写程序的效率低下，并且不能保证程序有好的质量。

为了能够更方便地编写程序，人们用一些符号和简单的语法来表示机器指令，这就是汇编语言。例如，“101110001110100000000011”用汇编语言表示就是“mov AX,1000”，该指令的功能是“将 1000 送入寄存器 AX 中”，是不是清楚多了？但是 CPU 并不能识别汇编语言，因此需要一个“翻译”程序将汇编语言翻译成机器语言，我们把这种将汇编语言翻译成机器语言的程序叫做“汇编器”。汇编语言与机器语言的指令是一一对应的，所以，除了提高了一些可读性，汇编语言并没有从根本上改变机器语言的特点。可以说，汇编语言是面向机器语言的。当然，汇编语言也仍然具备机器语言的优点。许多大型系统（例如操作系统）的核心部分都是用汇编语言编写的，因为这部分工作直接和硬件打交道，需要很高的效率。

那么，有没有办法真正提高程序的可读性、可维护性和可移植性呢？答案是肯定的，就是使用高级语言。

2. 高级语言

高级语言是一种比较接近自然语言和数学语言的程序设计语言。高级语言的出现大大提高了程序员的工作效率，降低了程序设计的难度，并改善了程序的质量。用高级语言编写的程序看起

来更像是英语，很容易读懂，不但使程序具备良好的可读性和可维护性，而且使更多的人掌握了程序设计方法，从而使计算机技术得到迅速的应用和普及。

例如，语句段

```
if(a>b)
   c=a;
else
   c=b;
```

表示的是“如果 a 大于 b，则 c=a，否则 c=b”。是不是很容易理解？当然，这里的“=”与数学语言中的等号是有根本的区别的，我们将在介绍 C 语言的运算符时详细地加以讨论。

另外，用高级语言编写的程序还具有很高的可移植性。从高级语言到机器语言要经过编译程序进行“翻译”，而高级语言几乎为每一种机器都创建了各自的编译程序，从而可以将用高级语言编写的程序几乎不加修改地运行在不同的计算机平台上。

编译程序分为两种，一种是解释系统，另一种是编译系统。解释系统是对高级语言编写的程序翻译一句执行一句；而编译系统是将高级语言编写的程序文件全部翻译成机器语言，生成可执行文件以后再执行。高级语言几乎在每一种机器上都有自己的编译程序。C 语言的编译程序属于编译系统。

1.2　C 语言发展概述和主要特点

1.2.1　C 语言的发展历史

C 语言与 UNIX 操作系统有着密切的关系，它的发明者是 Dennis Ritchie。Dennis Ritchie 开发 C 语言的主要目的是为了更好地描述 UNIX 操作系统。

1969 年，美国贝尔实验室的 Ken Thompson 在一台报废的 DEC PDP-7 上做了一些程序来辅助软件开发。

1969—1972 年，Ken Thompson 与 Dennis Ritchie 合作，用了不到四年的时间就把这些程序发展为一个操作系统——UNIX。

在此期间，早期的 UNIX 是用汇编语言写的，我们已经在前面谈到了汇编语言的缺点，Thompson 为了摆脱汇编语言的困扰，在 1970 年决定开发一种高级语言以便更有效地描述 UNIX，他以 BCPL 语言为基础开发了一种新的语言——B 语言。但 B 语言缺乏丰富的数据类型，又以字长编址，有一定的缺陷，因而未能流行起来。为了改进 B 语言，从 1971 年开始，Ritchie 用了一年左右的时间，在 B 语言的基础上加入了丰富的数据类型和强有力的数据结构，从而形成了 C 语言。之所以命名为 C 是因为：BCPL 是 B 语言的先驱，BCPL 这串字符中 C 字符在 B 字符的后面；同时，按 26 个字母的顺序（A、B、C、D、……），B 字符后面的也是 C。

1978 年，Brian W. Kernighan、Ken Thompson 与 Dennis Ritchie 三人合作，编写了著名的 *The C Programming Language*，该书介绍的 C 语言被称为标准 C。

而后，C 语言由于其本身的优点，先后被移植到各种计算机平台上，得到了广泛的使用。同时，也出现了很多编译系统版本。

1983 年，美国国家标准化协会（ANSI）建立了一个委员会，着手制定 ANSI 的标准 C。