高等教育质量工程信息技术系列示范教材

新概念
C程序设计大学教程

张基温 编著

清华大学出版社

内 容 简 介

本书是一本基于能力培养体系的C语言程序设计教材。本书按照作者提出的“提出问题、分析问题—编写程序、语法说明—程序测试、结果分析”的思路，并按照“前期以培养解题思路为主，语法知识够用就行；后期补充必要的语法细节”的原则编写，旨在引导读者在逻辑思维能力、语法应用能力和程序测试能力3个方面同步提高。

全书分为3篇。第1篇从几个经典问题入手，将读者带入穷举、迭代、递归、随机模拟、时间步长、事件步长等基本逻辑思维训练之中，并相对集中地融入基本语法，为初学者奠定程序设计的基本知识和能力。第2篇通过数组、结构体和指针3种构造数据类型以及常用算法设计策略的介绍，使读者的程序设计能力上升到“数据结构 + 算法 = 程序”的水平。第3篇对C语言重点语法进行总结、提升和拓展，使读者在发挥C语言优势方面得到提升。

本书结构新颖、概念准确，鱼渔并重、和木皆宜，例题经典、习题丰富、题型全面，适应面宽、注重效果，适合教学、兼顾自学，与教育部计算机科学与技术教学指导委员会推荐的《高等学校计算机科学与技术专业：公共核心知识体系与课程》中关于程序设计课程的要求一致，可以作为高等学校各专业的新一代程序设计课程教材，也可供从事程序设计相关领域的人员自学或参考。

图书在版编目（CIP）数据

新概念C程序设计大学教程 / 张基温编著. —北京：清华大学出版社，2012.6
（高等教育质量工程信息技术系列示范教材）
ISBN 978-7-302-28312-6

Ⅰ. ①新… Ⅱ. ①张… 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（2012）第044163号

责任编辑：白立军　李玮琪
封面设计：常雪影
责任校对：白　蕾
责任印制：王静怡

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　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：北京密云胶印厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：21.75　　**字　　数**：513千字
版　　次：2012年6月第1版　　**印　　次**：2012年6月第1次印刷
印　　数：1～3000
定　　价：35.00元

产品编号：045868-01

前　言

（一）

在信息时代，程序设计会被人称为计算机以及相关专业的看家本领，然而不仅如此，在程序设计中所蕴涵的逻辑，是解决所有领域复杂问题的根基，同时在程序设计课程中所进行的逻辑思维训练，所有人都可从中获益。但是，尽管程序设计课程已经开设几十年了，教学效果却很不尽如人意。

程序设计涉及逻辑思维、语言和方法3方面内容。然而，早期的程序设计课程仅仅是语言手册的改编。因此，从20世纪80年代，就开始着手改变这种状况，于1985年出版的我的第一本著作——《BASIC程序设计》中就在原来的语法体系中引入典型算法和软件工程思想。以后受CIT考试的启发，在为NIT（国家信息技术考试）编写的《程序设计（C语言）》（清华大学出版社，1999）一书中，把程序测试引入到程序设计教材中。

之后，几家出版社先后约稿，出版了《新概念C语言程序设计》（中国铁道出版社，2003）、《C语言程序设计案例教程》（清华大学出版社，2004）、《新概念C程序设计教程》（南京大学出版社，2007）、《新概念C语言教程》（中国电力出版社，2011）。在这些教材中，逐步形成并完善了按照内容体系的程序设计教材。所以将它们以“新概念”命名，是想表明这种全新理念的程序设计体系。让我欣慰的是，随着这几本书的不断改进，类似的书也陆续问世，品种不断增加，说明面向问题，按照“问题分析—设计代码—语法说明”线索组织程序设计教学的理念日益被广泛接受。

（二）

本书应清华大学出版社之邀而撰写，它是对前几本教材的进一步完善与改进。全书分为3篇。第1篇由5个单元组成，在这5个单元中，以几个经典问题为载体，以穷举、迭代、递归、随机模拟、时间步长、事件步长等基本算法为主线，将设计思路、程序测试方法和C语言基本语法知识融于其中。第2篇由4个单元组成。第6～8单元分别介绍数组、结构体和指针这3种支持程序数据结构的重要类型，使读者可以初步领略数据结构对于程序设计的重要性，第9单元介绍几种常用的算法设计策略。通过这4个单元，使“数据结构 + 算法 = 程序”的思想在读者心中扎根。前面两篇，按照“训练解题思路为主，语法够用就行”的原则编写。第3篇用6个单元补充了一些重要的语法细节，使读者能在前两篇初步掌握了程序设计的基本方法的基础上，将C语言程序设计的学习引向深入。这样不仅建立了一种全新的内容体系，将应试教育向能力培养方面做了较大幅度的转变，同时也与教育部计算机科学与技术教学指导委员会推荐的《高等学校计算机科学与技术专业：公共核心知识体系与课程》中关于程序设计课程的要求一致，读者可以按照自己的专业定位选择其中的有关内容。

（三）

为了能有的放矢地进行训练，本书以二级节为单位给出习题。习题分为4个栏目：概念辨析、代码解析、探索验证和开发练习。

“概念辨析”主要提供了一些选择题和判断题，旨在提高读者对基础语法知识的认知。

“代码解析”包括指出程序（或代码段）执行结果、改错和填空，旨在提高读者的代码阅读能力。因为读程序也是程序设计的一种基本训练。

“探索验证”主要是用于提示或者指导学习者如何通过自己上机验证来提高掌握语法细节的能力。除了这个栏目中的习题外，学习者最好也能通过设计程序验证自己对于概念辨析栏目中的习题的判断是否正确。

“开发练习”是一种综合练习，应当要求学习者写出开发文档。内容主要包括问题（算法）分析、代码设计、测试用例设计、测试及调试结果分析等几个部分，重点应当放在问题分析、代码设计和测试用例的设计上。要把这些工作都做好，再上机调试、测试，不要什么还没有设计出来就上机。

（四）

在本书即将出版之际，由衷地感谢在本书写作过程中参与收集资料、程序调试以及校阅工作的姚威、张秋菊、文明瑶、杜勇、丁群、朱莎、史林娟、张展为、张有明。同时，也殷切地期待着广大读者和同仁的批评和建议。让我们共同把程序设计课程的改革做得更有实效。

张基温

2012 年 2 月

目　录

第 1 篇　C 语言程序设计初步

第 2 篇　数据结构＋算法

第3篇　深入学习C语言

第1篇　C语言程序设计初步

计算机是人类历史上迄今为止最为神奇的一种机器。它几乎“无所不能”地在人类社会的每个领域大显身手。这种神奇之所在主要来自它的程序。因为计算机的工作完全是由程序控制的，程序安排计算机做什么，计算机才能做什么。离开了程序，计算机就只剩下一堆金属和塑料。

程序是由人编写的。为了编写程序，人要先理清自己的解题思路，建立一个问题的模型，并用计算机能直接或间接理解的符号系统描述出来。这个过程可以简单地记为

程序＝模型＋表现

计算机能直接或间接理解的、用于描述解题思路的符号系统称为计算机程序设计语言，简称计算机语言或程序设计语言。本书介绍的C语言，是目前在计算机语言市场上占据统治地位的一种计算机程序设计语言。

学习程序设计，主要涉及两个方面：培养解题思路和掌握某种计算机程序设计语言的语法规则。其中解题思路的培养尤为重要。在计算机科学中，把问题的求解思路称为算法(algorithms)。算法是程序的灵魂。相对于算法的培养，程序设计语言语法的学习就简单多了。这一篇通过几个实例，介绍几种程序中常用的算法和用C语言表现这些算法的基本方法，为进一步学习奠定基础。

C语言的来历

C语言的来历要从ALGOL(也称A语言)说起，ALGOL是首批高级程序设计语言之一。1963年剑桥大学基于ALGOL 60提出了CPL (Combined Programming Language)。1967年剑桥大学的Matin Richards对CPL语言进行了简化，形成了BCPL语言。1970年贝尔实验室的Ken Thompson将BCPL“煮”了一下，提炼出其精华，将之命名为B语言。他还用B语言写了第一个UNIX操作系统。1973年贝尔实验室的Dennis M. Ritchie又把B语言“煮”了一次，在此基础上设计出了一种新的语言，将之称为C语言。1977年Ritchie发表了不依赖于具体机器系统的C语言编译文本——可移植的C语言编译程序。1978年Brian W. Kernighian和Dennis M. Ritchie(如图0-1所示)出版了名著《The C Programming Language》，把C语言推向流行最广泛的高级程序设计语言宝座。

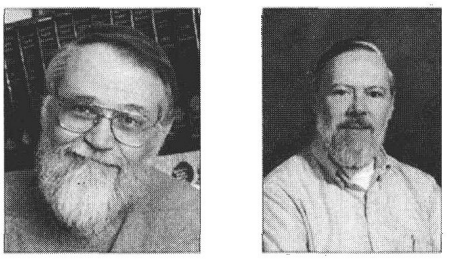

图0-1　Brian W. Kernighian和Dennis M. Ritchie

随着C语言的流行，特别是在1988年后，随着微型计算机的日益普及，出现了许多C语言版本，导致了应用上的困难。为了改变这种情况，美国国家标准学会(American National Standands Institute, ANSI)为C语言制定了一套ANSI标准，成为现行的C语言标准。但是，任何标准都不可能包罗万象，总有考虑不周之处。而且C语言的标准也在不断调整。所以，现在实际应用的C语言系统，因版本不同，还在细微处存在一些区别。

第1单元　简单的C语言程序

本单元通过一个简单的实例，介绍C语言程序的基本结构。

1.1　两个整数相加

1.1.1　两个整数常数相加的C语言程序

【代码1-1】　一个用于计算2+3的C语言程序。

```
int main (void) {
    2+3;
    return 0;
}
```

说明：本例是一个最简单的C语言程序。图1-1所示为这种最简单的C语言程序的结构——它由一个名为main的函数组成。

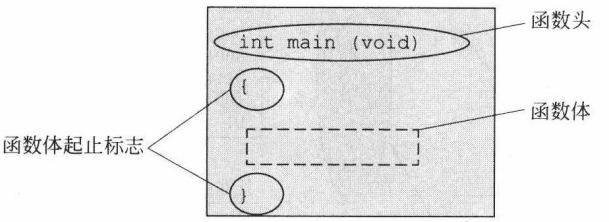

图1-1　C语言主函数的基本框架

main函数由一个函数头和一个函数体组成。除非有特殊用途，否则就认为main函数的函数头是一个固定的形式：int main(void)。函数体是用一对花括号括起来的一组声明(declaration)或语句。声明是向编译器所作的关于数据的说明。语句用于向计算机发出操作指令。C语言规定，每个声明和语句都以分号结束。在代码1-1中，包含了两条语句："2+3;"和"return 0;"。

2+3称为一个C语言表达式。C语言表达式是用操作数和操作符组成的合乎C语言语法的式子。在2+3这个表达式中，2和3称为操作数；"+"称为操作符，准确地说，是一个算术操作符。C语言的操作符有算术操作符、关系操作符、逻辑操作符、赋值操作符等十几种类型。其中，算术操作符有+(加)、-(减)、*(乘)、/(除)、%(模，即整除取余)。例如，19%7的结果是5。

1.1.2　C语言程序的编译与连接

C语言不是计算机CPU(中央处理器)可以直接理解的程序设计语言。因为C语言是

一种靠近英语的程序设计语言，使用这样的语言编写程序，可以使程序员把精力集中在解题思路的设计上。而计算机 CPU 所能直接理解的是用 0、1 码描述的指令系统。为此，要计算机 CPU 理解并执行一个 C 语言程序，就要将这个 C 语言程序转换——翻译成所在计算机 CPU 机器语言程序。这个过程称为编译(compile)。为了区别程序员编写的程序和经过编译的程序，将前者称为源程序代码或源程序，将后者称为目标程序代码或目标程序。实现编译功能的程序称为编译器(compiler)。

但是，目标程序还不是可以执行的程序。因为，随着计算机所求解的问题越来越复杂，计算机程序的规模也越来越庞大。为了便于发现错误和纠正程序中的错误，采用了分块编写程序的方法。一个程序在编写过程中形成了多个模块，并且一些是程序员自己编写的，有些可以使用别人或开发商提供的。要让系统正确地执行一个程序，必须将这些经过编译的程序资源同主函数连接起来，这个过程称为链接(或连接)。执行这个任务的程序称为连接器。只有经过链接的程序才可以成为可执行程序。

图 1-2 描述了一个 C 语言程序的编译和链接过程。

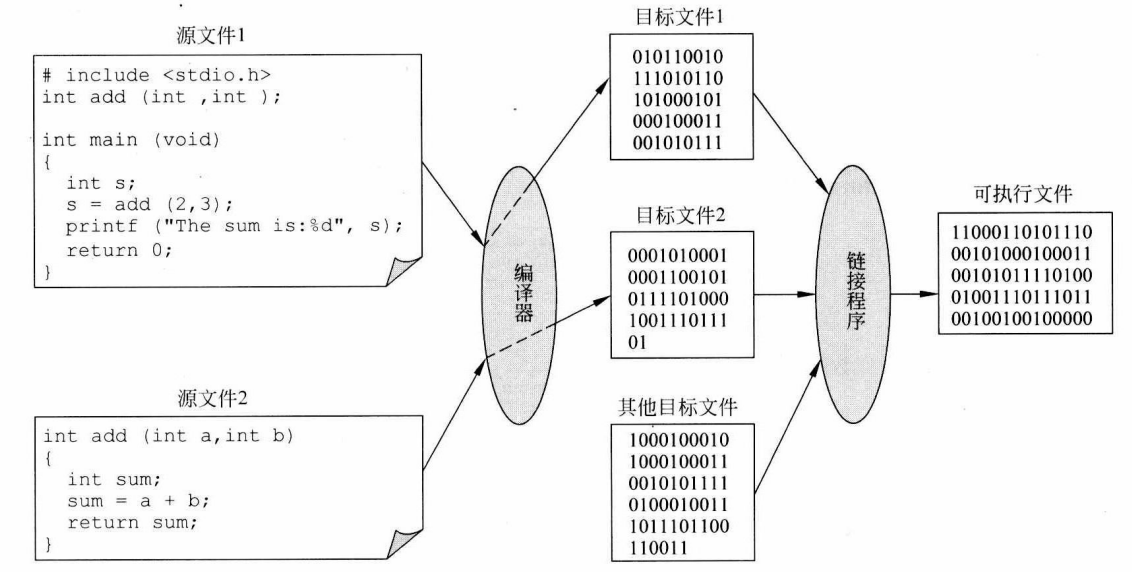

图 1-2　C 语言程序的编译和链接过程示意图

一个完整的高级语言程序的开发过程，就是从问题出发，编写出高级语言程序，再编译链接得到可执行程序的过程。图 1-3 显示了这个过程。

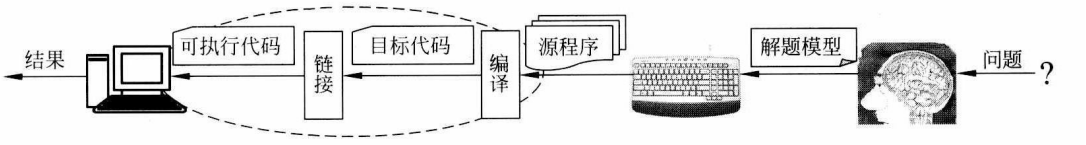

图 1-3　高级语言程序开发的过程

1.1.3　带有输出操作的 C 程序

如果在计算机上运行代码 1-1，计算机上什么也不会出现，即得不到任何结果。什么原

因呢？不是计算机没有进行计算，而是这个程序没有输出功能，犹如“哑巴吃饺子——心中有数”却没有表达出来。因此，一个程序必须有输出功能，把运行的状况和结果告诉用户。

【代码 1-2】 带有输出操作的计算 2＋3 的 C 语言程序。

```
#include <stdio.h>
int main (void) {
    printf ("%d",2+3);
    return 0;
}
```

执行这个程序，计算机屏幕上显示如下：

```
5
```

有以下几点说明。

(1) 代码 1-2 的程序结构如图 1-4 所示。它由两个函数组成：main()函数和 printf()函数。在函数 main()中，用语句“printf(“%d”,2＋3);”调用函数 printf()的功能。当操作系统接受人的命令开始执行这个程序时，首先从函数 main()开始执行，当需要 printf()函数的功能时，就调用函数 printf()，把执行流程转交到函数 printf()中，执行完函数 printf()，再返回到函数 main()中。

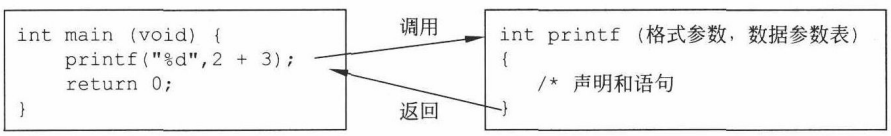

图 1-4　代码 1-2 的程序结构

一个 C 语言程序的一般情况如图 1-5 所示。即由一个或多个函数组成，但必须有且只能有一个函数 main()。其他函数都是由函数 main()直接或间接调用的。所以，把函数 main()称为主函数。或者说，执行一个 C 语言程序，就是从主函数开始到主函数结束的执行过程。

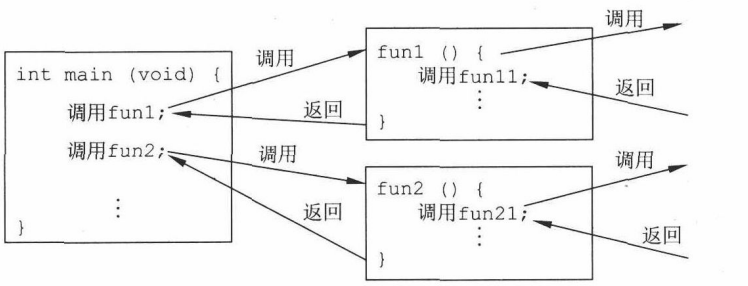

图 1-5　C 语言程序的一般结构

(2) 代码 1-2 有两个函数，可是为什么看不见函数 printf()的定义的呢？这正是 C 语言的一个优点。C 语言追求精干、高效和灵活，它没有设置专门的输入和输出语句，输入和输出采用一些库函数实现。printf()就是最常用的一个用于格式化输出的库函数。即这个函数定义在系统的函数库中。程序员要使用库函数，有两个要求：第一是要正确地写出函数

printf()的调用语句。第二是要让编译器能根据函数的基本信息(称函数原型)来检查这个函数的调用语句是否正确并找到这个函数的定义。这就是在代码 1-2 的开头写了一行“#include <stdio.h>”的原因。因为在头文件 stdio.h 中有 printf()的函数原型信息。注意,#include 是一条编译预处理命令(不是语句,不用分号结尾),用于将一个文件包含到当前程序中。图 1-6 所示为对一个源代码文件进行文件包含预处理的示意图。

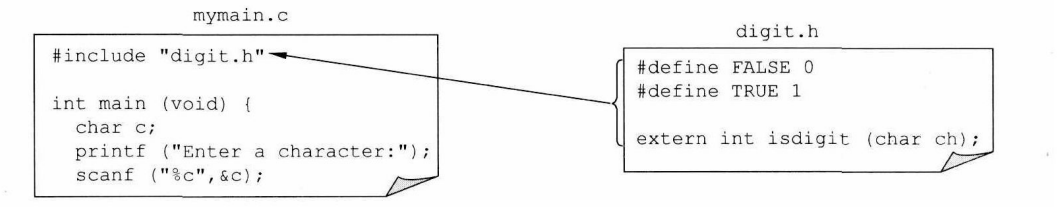

图 1-6 一个源代码文件中的文件包含预处理示意图

(3) 如图 1-7 所示,printf()的参数分为两部分:前面一部分被括在一对双撇号中,称为格式参数。其中的%d 称为格式字段,%表明后面的字符 d 不是普通的字符,而是一个用来说明后面的数据项要用十进制整型格式输出的符号,称为整型格式字符。后面部分参数称为数据参数,如本例中的 2+3。两部分合起来就表示:把表达式 2+3 的值用整型格式显示出来。

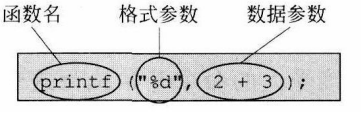

图 1-7 函数 printf()的调用表达式

用一个 printf()函数可以输出多个内容表达式的值。

【代码 1-3】 用 printf()函数输出多个内容表达式的值。

```
#include <stdio.h>
int main (void) {
    printf ("%d+%d=%d",1+1,1+2,2+3);
    return 0;
}
```

代替

显示

执行这个程序,会在计算机屏幕上显示如下:

```
2+3=5
```

计算机执行这个 printf()函数时,首先分别计算后面的 3 个内容表达式的值,再分别按照顺序用计算出的 3 个值代替格式参数中的格式字段“%d”,最后将形成的一串字符显示出来。可以看出,原来格式字符串中的+、=都原样显示出来了,只有格式字段被用后面表达式的值替换了。显然,需要几个输出项,就需要在格式字符串中使用几个用%开始的格式字段。

习 题 1.1

一、概念辨析

选择正确的答案。

(1) 下面叙述中,错误的是(　　)。

A. 一个 C 语言程序至少要有一个主函数