普通高等教育“十一五”国家级规划教材

张光妲 主编
陶佰睿 主审

C语言程序设计教程

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

张光妲 主编
李君 尚晓丽 吕洪柱 于晓敏 副主编

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

本书是根据教育部高等学校计算机科学与技术教学指导委员会非计算机专业计算机基础课程教学指导分委员会提出的《非计算机专业计算机基础课程教学基本要求》和《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求(试行)》中提出的要求,按照以计算思维为导向的分类、分层次组织教学的思路,并根据C语言的特点和初学者的认知规律,结合高等学校计算机语言课程改革的要求而编写的。本书在内容组织上通过例题来介绍C语言的一些基本概念,让读者在做中学,在编程中体会,避免枯燥基础知识的简单介绍过程。通过合理布局,适当地对内容进行了删减。全书共分10章,每章均配有教学课件和精心设计的习题。

本书可作为高等院校C语言程序设计课程的教材,也可供广大计算机爱好者学习参考。

图书在版编目(CIP)数据

C语言程序设计教程/张光妲主编. --北京:清华大学出版社,2014
21世纪计算机科学与技术实践型教程
ISBN 978-7-302-35263-1

Ⅰ. ①C…　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(2014)第016197号

责任编辑:袁勤勇　徐跃进
封面设计:何凤霞
责任校对:李建庄
责任印制:沈　露

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
课件下载:http://www.tup.com.cn,010-62795954
印 装 者:北京鑫海金澳胶印有限公司
经　　销:全国新华书店
开　　本:185mm×260mm　　**印　　张**:18.75　　**字　　数**:463千字
版　　次:2014年3月第1版　　**印　　次**:2014年3月第1次印刷
印　　数:1～2000
定　　价:32.00元

产品编号:057659-01

《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。

制订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，而且也学会了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语言多年来一直作为高等学校计算机程序设计的入门课程之一，具有功能丰富、应用灵活、执行效率高并能直接对硬件操作等特点，既可以作为系统软件的描述语言，也可以用来开发应用软件。通过学习C语言，学生不仅能够掌握程序设计的基本思想，培养计算思维能力，也可为今后学习Java、C++等语言打下良好的基础。

本书是根据教育部高等学校计算机科学与技术教学指导委员会非计算机专业计算机基础课程教学指导分委员会提出的《非计算机专业计算机基础课程教学基本要求》和《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求(试行)》中提出的要求，按照以计算思维为导向的分类、分层次组织教学的思路，由多年从事一线任课教师根据教学改革经验和课程特点编写而成。本书的多名作者常年担任高级语言类课程的理论和实验教学任务，出版过多部相关教材，而且本书还是在教研项目“基于计算思维的计算机基础教学改革与探讨”的研究基础上进行编写的。

全书共分10章，主要包括C语言概述，数据类型、运算符与表达式，顺序程序设计，选择结构程序，循环控制，数组，函数，结构体与共用体，指针与链表以及文件。

第1章C语言概述，介绍程序与程序设计的基本概念，C语言的发展简介、特点、简单的C程序，算法的概念、特点及如何用流程图来表示算法。

第2章数据类型、运算符与表达式，介绍C语言的数据类型、各种数据类型的变量与常量、运算符及其对应的表达式。

第3章顺序程序设计，介绍顺序结构及C语言的格式输入输出。

第4章选择结构程序设计，介绍选择结构程序中的单分支语句、双分支语句、switch语句，并对这几种分支语句进行比较。

第5章循环控制，介绍C语言中提供的while语句、do-while语句和for语句，并对这三种循环进行了分析和对比。

第6章数组，介绍一维数组、二维数组和字符数组的定义、引用以及初始化，介绍了数组的典型实例和常用的字符串处理函数。

第7章函数，介绍函数的定义、调用、返回值、参数传递，函数的嵌套调用和递归调用以及变量的作用域与存储类别。

第8章结构体与共用体，介绍结构体和共用体的定义，变量的使用方法以及使用时二者的区别。

第9章指针与链表，介绍指针变量、指针与数组的关系、指针与函数的关系、指针与结

构体变量的关系及链表的实现。

第10章文件，介绍文件、文件指针变量以及对文件的操作。

本书整体结构编排合理，在内容组织上通过例题介绍C语言的一些基本概念，让读者在做中学，在编程中体会，避免枯燥基础知识的简单介绍过程。通过本书的学习，学生能够掌握高级语言程序设计的基本思想和常见问题的算法描述，并可以自行编写程序加以实现。

本书由张光姐、李君、尚晓丽、吕洪柱、于晓敏编著，其中张光姐编写了第8章、第9章、第10章，并负责全书的统稿和审定工作；李君编写了第1章、第6章、第7章，尚晓丽编写了第4章、第5章，吕洪柱编写了第2章、第3章(除3.6节)，于晓敏编写了3.6节；陶佰睿教授审阅了全书，并提出了许多宝贵的意见。

本书在编写过程中得到了清华大学出版社和教学同行的大力支持和帮助，在此表示衷心的感谢。由于作者水平有限，书中难免有不妥之处，敬请读者和专家批评、指正。

编著者

2014年1月

目　　录

第1章　C语言概述

1.1　程序与程序设计

计算机所做的工作就是按指定的步骤执行一系列操作，以完成某一特定的任务。因此要计算机为我们做事，就必须事先设计一些操作步骤，并用计算机语言写成程序。通俗地讲，程序是向计算机发出的一个个操作“命令”的集合，告诉计算机如何完成一个具体的任务；专业地说，程序就是为实现特定目标或解决特定问题而利用计算机语言编写的命令序列的集合。

一个程序应包括对数据的描述和对操作的描述两部分内容。其中，对数据的描述是指数据的类型和组织形式，即数据结构；对操作的描述是指求解问题的步骤，即算法。著名计算机科学家 Nikiklaus Wirth 提出了公式：

数据结构＋算法＝程序

这个公式说明构成一个程序必须要有两大组成要素：数据结构和算法。此外，一个程序还应采用程序设计方法来设计，最终使用计算机语言表示出来。

1965 年荷兰科学家 E. W. Dijikstra 提出了结构化程序设计的思想，其实质是控制编程中的复杂性，以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，使完成每一个模块的工作变得简单而明确。结构化程序设计可以把完成某一个任务的复杂过程分解为若干个子过程，子过程再分解，一直到某些相对简单的过程，从而为设计一些较大的软件打下良好的基础。

结构化程序设计的基本思想是采用“自顶向下、逐步细化、模块化设计、结构化编码”的程序设计方法。

“自顶向下、逐步细化”的程序设计方法就是指从问题本身开始，经过逐步细化，将解决问题的步骤分解为由基本程序结构模块组成的结构化程序。具体来说，“自顶向下”就是要求在程序设计时，应先考虑总体，后考虑细节；先考虑全局目标，后考虑局部目标。“逐步细化”则要求对于一个复杂的问题，应设计出一些子目标作为过渡。在程序设计中，往往将要解决的总目标分解为子目标，或者再进一步分解为具体的小目标，其中每一个小目标都称为一个模块。划分子模块时应注意模块的独立性，即使用一个模块来完成一项功能，耦合性越少越好。

在结构化程序设计中，对于规模稍大的程序，可以按照所要实现的功能来划分模块，每一个模块都可以看作是一个具有单输入、单输出的独立程序，所设计的大规模的程序实

际上是由一些具有相对独立功能、结构清晰、容易理解的小程序模块组成的。在利用C语言进行程序设计时，可以将这些小程序用函数定义成“模块”，即将程序模块化。

结构化程序设计中所使用的基本程序结构有三种：顺序结构、选择结构和循环结构，任何复杂的算法，都可以由这三种基本结构组成。利用这三种结构所对应的语句可以实现结构化编码，在结构化编码中，所有的语句都有可能被执行，程序中不能出现“死循环”。

1. 顺序结构

顺序结构是一种最简单的结构，它是指程序流从上向下顺序执行，各个模块都仅执行一次。

2. 选择结构

选择结构又叫分支结构，是指程序的流程发生分支，根据一定的条件从两个或更多的模块中选择一个模块来执行，而且该模块只能执行一次，其他模块则不会被执行。

3. 循环结构

循环结构是指有条件地重复执行某个模块，它可以简化程序的难度，将大工作量拆分成小工作量，并对小工作量进行重复操作，这种方法充分利用了计算机运算速度快、自动化的优点。

1.2 C语言简介

1.2.1 C语言的发展

C语言的原型是ALGOL 60语言(也称为A语言)，1963年，剑桥大学将其发展成为CPL语言。1967年，剑桥大学的Matin Richards对CPL语言进行了简化，产生了BCPL语言。1970年，美国贝尔实验室的Ken Thompson将BCPL进行了修改，为其起名为“B语言”，并用它写了第一个UNIX操作系统。1973年，美国贝尔实验室的D. M. Ritchie在BCPL和B语言的基础上最终设计出了一种新的语言，这就是C语言，随后不久，UNIX的内核和应用程序全部用C语言改写。1977年出现了可移植的C语言编译程序，使得用C语言编写的UNIX系统可以在各种计算机上使用。随着UNIX的广泛应用，C语言得到了普及和推广，并最终独立于UNIX而成为世界上应用最广泛的计算机语言之一。

1978年，Dennis Ritchie和Brian Kernighan合作推出了“The C Programming Language”的第一版(简称为K&R)，人们称之为K&R C。由于在K&R中并没有定义一个统一的标准，1983年，美国国家标准协会(ANSI)公布了第一个C语言标准草案，即ANSI C。1989年，草案被ANSI正式通过成为美国国家标准，被称为C89标准。1990年国际标准化组织(ISO)接纳了ANSI C，并做了一些小的修改，也就形成了ISO C(又称为C90)。之后又进行了几次修改，最近一次C规范修订在1999年制定，即常称的C99规范。在目前微型机上使用的各种C编译器都提供了对C89(C90)的完整支持，对C99还只提供了部分支持，还有一部分提供了对某些K&R C风格的支持。

1.2.2 C语言的特点

C语言是一种面向过程的程序设计语言，它既具有高级语言的特点，又有汇编语言的特点。C语言之所以被广泛使用，正是归功于其自身所拥有的特点，C语言的主要特点如下：

(1) 语言简洁、紧凑，使用灵活、方便。C语言只有32个关键字，9种控制语句，程序书写自由。

(2) 运算符极其丰富。C语言共有34个运算符。C语言把括号、赋值、强制类型转换等都作为运算符处理，从而使运算类型极其丰富，可以实现在其他高级语言中难以实现的运算。

(3) 数据类型丰富。C语言具有整型、实型、字符型、数组、指针、结构体、共用体等多种数据类型，能够实现各种复杂的数据类型的运算。特别是C语言的指针类型，功能强大、灵活方便。

(4) C语言是结构化、模块化语言。C语言具有多种循环、条件语句控制程序流向，从而使程序完全结构化。C语言是函数式语言，以函数作为程序模块单位，使得程序结构清晰、可读性好、易于调试。

(5) 语法限制不太严格，程序设计自由度大。例如，对数组下标越界不作检查，对变量的类型约束不严格等。C语言放宽了语法检查，使得程序编写有了较大的自由度，但也同时要求编程人员要自己检查程序，保证程序正确，而不要过分依赖编译程序。

(6) C语言允许直接访问物理地址，可以直接操纵硬件。C语言能够对位、字节和地址进行操作，可用来编写系统软件。

(7) 生成的目标代码质量高，程序执行效率高。用C语言编写的程序，其生成的目标代码通常只比汇编程序生成的目标代码效率低10%～20%。

(8) 与汇编语言相比，用C语言编写的程序可移植性好。用C语言编写的程序，基本上不做修改就可以直接应用于各种计算机和操作系统。

1.3 简单的C程序

下面通过几个例子，分析C程序的基本结构特点。

例1.1 在计算机屏幕上输出“Hello World!”。

```
#include<stdio.h>                    /*编译预处理命令*/
main()                               /*主函数*/
{
    printf("Hello World!\n");        /*输出“Hello World!”*/
}
```

该程序的第一行是一条编译预处理命令，由于该程序需要使用标准库函数中的printf函数，因此需要在程序的开头将定义该函数的头文件stdio.h包含到程序中来。在

该程序中，main 是函数的名字，表示“主函数”，任何一个 C 语言程序都必须有且只能有一个主函数(即 main 函数)。main 函数名的后面有一对小括号，函数体由大括号{}括起来，这些括号都必须成对出现。在该程序的函数体内只有一条语句，该语句利用 printf 函数输出字符串“Hello World!”，其中\n 表示在输出“Hello World!”后将光标移到下一行，语句结束用分号表示。此外，程序中的/* */为注释，注释只是写给程序员看的，用来向用户提示或解释程序的意义，在程序编译时，不对注释作任何处理，注释可出现在程序中的任何位置。

例 1.2 计算两个数的和。

```
#include<stdio.h>
main()                              /*主函数*/
{
    int a,b,sum;                    /*定义变量a、b、sum*/
    scanf("%d %d",&a,&b);           /*输入变量a和b的值*/
    sum=add(a,b);                   /*调用add函数,将获得的返回值赋给变量sum*/
    printf("sum=%d\n",sum);         /*输出变量sum的值*/
}
add(int x,int y)                    /*add函数,其中x和y是形式参数*/
{
    int z;
    z=x+y;                          /*将形参x与形参y的和赋给变量z*/
    return(z);                      /*返回变量z的值*/
}
```

在该程序中，一共有两个函数，分别是 main 函数和 add 函数。在主函数 main 中，首先定义整型变量 a、b、sum，利用 scanf 函数让用户为变量 a、b 输入值，然后以 a、b 作为实参调用 add 函数，在调用 add 函数时，将实参 a、b 的值分别传递给形参 x、y，然后将 x+y 的和赋给变量 z，并返回给主调函数 main，在 main 函数中再将获得的返回值赋给变量 sum，最后利用 printf 函数输出 sum 的值。

该程序中使用了 scanf 函数和 printf 函数来实现数据的输入输出，涉及了函数调用、参数传递等内容，在此只做简单解释，这部分内容将在后面的章节中详细介绍。

通过以上介绍的例子，可以看出 C 程序的语法格式与结构特点如下：

(1) C 程序是由函数构成的。一个 C 程序可以由一个或多个函数组成，其中必须有且只能有一个主函数 main。

(2) 函数由函数说明和函数体组成。函数说明部分包括函数的名称、函数的返回值类型、函数的参数及类型；函数体由一对大括号括起来的若干条声明语句和执行语句构成。

(3) 无论一个 C 程序中有多少个函数，都必须从主函数 main 开始执行，最后在 main 中结束，其他函数都是通过函数调用得以执行。

(4) C 程序的书写格式较自由，一行内可以写多条语句，一条语句也可以分写在多行上。

(5) C语言程序中可以有预处理命令(如#include命令等),预处理命令单独占一行书写,末尾不加分号,是在源程序编译前做一些处理工作。

(6) 每条语句和数据声明的最后必须有一个分号,分号是C程序中语句的结束标志。

(7) C语言中的数据输入输出功能可以通过scanf和printf等函数实现。

(8) 关键字、标识符之间必须用至少一个空格加以分隔。

(9) C程序习惯采用小写字母书写,并区分大小写。

(10) 可以用/*……*/对C程序中的任何部分进行注释,适当的注释可以提高程序的可读性。

1.4 算 法

1.4.1 算法概述

我们在日常生活中处理事情时往往都有一定的方法和步骤,例如,手洗一件衣服,需要把衣服放入盆中、接水、加入洗衣粉、搓洗、用清水投洗、拧干、晾晒等步骤。计算机处理某一问题的过程与我们日常处理事情的过程十分相似,都要按照一定的方法和步骤来处理,这些方法和步骤就是解决问题的算法。

简单来说,算法就是为解决某一问题而采取的方法和步骤,或者说是对解决一个问题的方法的精确描述。算法是能被机械地执行的动作或指令的有穷集合,其中每一条指令表示一个或多个操作。算法代表着用系统的方法描述解决问题的策略机制,规定了解决某一特定类型问题的一系列运算,是对解题方案的准确和完整的描述。制定一个算法,通常需要经过设计、确认、分析、编码、测试、调试、计时等阶段。计算机算法可以分为两类:数值运算算法和非数值运算算法,其中数值运算算法是用来求解数值的,而非数值运算算法通常用于事务管理领域。

一个算法应该同时具备以下主要特性。

(1) 有穷性:一个算法必须保证在执行有限个步骤之后结束,且每一步都可在有限的时间内完成。

(2) 确定性:算法中的每一个步骤都应有确切的含义,而不能是含糊的、模棱两可的。

(3) 可行性:一个算法必须是能行得通的,即算法中所描述的操作都可以通过执行有限次基本运算来实现。

(4) 有零个或多个输入:输入为算法指定了初始条件,其中零个输入是指算法本身已设定了初始条件。

(5) 有一个或多个输出:输出是与输入有某种特定关系的量,算法的实现是以得到计算结果为目的的,没有输出的算法毫无意义。

当算法中的运算在计算机上执行时,需要一定的存储空间来存放描述算法的程序和算法所需的数据,需要一定的时间来完成运算任务。通常,为解决某一问题而设计的算法并不唯一,不同的算法可能用不同的时间和空间来完成同样的任务,因此算法在某些方面