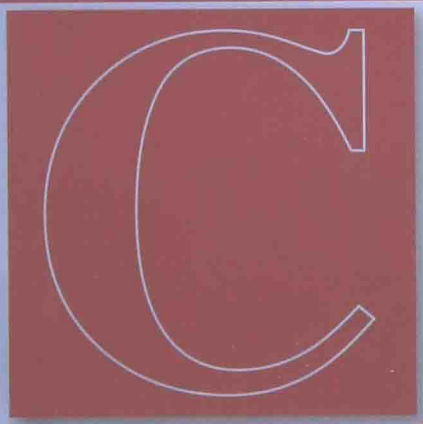
21世纪高等学校计算机基础实用规划教材

计算机程序设计基础教程——C语言

刘卫国 童键 主编



清华大学出版社

21世纪高等学校计算机基础实用规划教材

计算机程序设计基础教程——C语言

刘卫国 童键 主编

清华大学出版社
北京

内 容 简 介

本书遵循以计算思维能力培养为切入点的教学改革思路，以C语言作为实现工具，介绍程序设计的基础知识与基本方法。全书的主要内容有程序设计概述、程序的数据描述、顺序结构程序设计、选择结构程序设计、循环结构程序设计、函数与编译预处理、数组、指针、结构体、共用体与枚举、文件操作等。

在本书编写过程中，考虑到初学者的认知特点以及培养程序设计能力的教学要求，对C语言本身的语法规则做了适当处理和组织编排，突出C语言的重要概念和本质特点。全书以实际问题的求解过程为向导，突出从问题到算法，再到程序的一种思维过程，强调计算机求解问题的思路引导与程序设计思维方式的训练，重点放在程序设计的思想与方法上。

本书可作为高等学校计算机程序设计课程的教材，也可供参加各类计算机等级考试的读者以及社会各类计算机应用人员阅读参考。

图书在版编目(CIP)数据

计算机程序设计基础教程——C语言/刘卫国，童键主编. --北京：清华大学出版社，2014
21世纪高等学校计算机基础实用规划教材
ISBN 978-7-302-37002-4

Ⅰ. ①C…　Ⅱ. ①刘…　②童…　Ⅲ. ①C语言－程序设计－教材　Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2014)第139531号

责任编辑：魏江江　薛　阳
封面设计：常雪影
责任校对：李建庄
责任印制：李红英

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 刷 者：北京富博印刷有限公司
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：20　　**字　　数**：501千字
版　　次：2014年11月第1版　　**印　　次**：2014年11月第1次印刷
印　　数：1～2000
定　　价：36.00元

产品编号：029492-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程（简称‘质量工程’）”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程（群）加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域，以公共基础课为主、专业基础课为辅，横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

（1）面向多层次、多学科专业，强调计算机在各专业中的应用。教材内容坚持基本理论适度，反映各层次对基本理论和原理的需求，同时加强实践和应用环节。

（2）反映教学需要，促进教学发展。教材要适应多样化的教学需要，正确把握教学内容和课程体系的改革方向，在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生的知识、能力、素质协调发展创造条件。

（3）实施精品战略，突出重点，保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上；特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现教学质量和教学改革成果的教材。

（4）主张一纲多本，合理配套。基础课和专业基础课教材配套，同一门课程可以有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化，基本教材与辅助教材、教学参考书，文字教材与软件教材的关系，实现教材系列资源配套。

（5）依靠专家，择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建

设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主题。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21世纪高等学校计算机基础实用规划教材

联系人：魏江江 weijj@tup.tsinghua.edu.cn

前言

“计算机程序设计基础”是一门非常重要的计算机课程，其目的是介绍程序设计的基础知识，使学生掌握高级语言程序设计的基本思想、方法和技术，理解利用计算机解决实际问题的基本过程和思维规律，从而更好地培养学生的创新能力，为未来应用计算机进行科学研究与实际应用奠定坚实的基础。

近年来，计算机教育界提出，应将计算思维能力培养作为计算机教育的重要任务。计算思维(Computational Thinking)是指运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解的一系列思维活动。计算思维不仅反映了计算的原理，更重要的是体现了基于计算机的问题求解思路与方法。就课程性质而言，计算机程序设计基础课程最能够体现问题求解方法，是理解计算机工作过程的有效途径，也是计算思维能力培养的重要载体。因此，计算机程序设计基础课程的重要性不仅体现在一般意义上的程序设计能力的培养，而且体现在引导学生实现问题求解的思维方式的转换，即学生计算思维能力的培养。当然，要实现计算思维能力的培养不是一件容易的事，这也是程序设计教学改革的重要切入点。本教材正是按照这种改革理念，以实际问题的求解过程为向导，介绍程序设计的基础知识与基本方法，教材内容强调计算机求解问题的思路引导与程序设计思维方式的训练，重点放在程序设计的思想与方法上。

C语言是目前流行的程序设计语言之一，具有程序简洁、数据类型丰富、表达能力强、使用灵活、实用高效等优点，在当今软件开发领域有着广泛的应用，也是高等学校最常用的程序设计教学语言之一。诚然，当下C语言程序设计的书籍不胜枚举。作者分析了这些书籍，发现大致有两种处理方法，一种是按照语言的语法体系组织教材，先讲语法知识，再举例来说明这些语法的应用，这样做的好处是语言本身的语法体系完整系统，便于初学者学习掌握。语法知识实际上也是很重要的基本功，早些时候该课程的名称就叫“算法语言”。但人们担心，专注于语法，冲淡了程序设计能力的培养，于是就有另外一种教材组织模式，即按问题组织教材内容，先提出问题，再寻找解决办法，引出语法规则，这样做的好处是将学习的注意力放在解决问题的方法上，但显然程序语言的系统性没有了，初学者学习起来似乎也有困难。经过多年教学改革实践，我们认为，突出程序设计能力培养是十分必要的，这是计算思维能力培养的必然要求，但给学生完整的语言体系也是必要的。因此，如何处理好语法体系和求解问题方法的矛盾，是教材内容组织的关键问题。我们提出，在保持完整语法体系的前提下，给学生一个完整的解决问题的思路，这是解决问题的根本途径。为此，本书在编写过程中，力求体现4个方面的特点。

一是全书强调计算机问题求解的思路引导，突出从问题到算法，再到程序的一种思维过程。不是罗列现成的程序，而是讲清楚程序是怎么来的，怎样才能得到程序。各章的序言部分讲述不同的语言要素在问题求解中的作用，由此引出本章的内容。在讲程序实例时，先条理性地列出问题求解的基本步骤，再对基本步骤进行逐步细化，最后得到完整的算法。有些例子更

多地是从教学的角度设计的，这是应用的基础和前提，有些例子则具有很强的实际应用背景，可以更好地培养读者的应用开发能力。书中穿插介绍了递推法、迭代法、穷举法、试探法、递归法、分治法等算法设计策略，有利于读者掌握有关程序设计方法。在语言编译系统的选择上，本书使用 Visual C++ 6.0 作为上机环境，目的是让教材内容更加接近软件开发的实际需要，为读者进一步学习和应用 C++打下基础。

二是恰当取舍，突出 C 语言的本质特点和教学要求。全书用通俗易懂的叙述讲述 C 语言的重要概念，不求面面俱到。对于初学者不常用到的内容（如位运算）做了简化处理。教材也不过分死抠语言细节（如++和--运算符的副作用），引导读者在实践中去掌握语法规则。

三是全书的组织编排遵循循序渐进原则。教材前 6 章体现了基本程序设计能力的训练，第 1 章介绍程序设计的基础知识，建立起对 C 语言的初步认识；第 2 章介绍程序的数据描述，在这一章中并未罗列全部表达式，而是将相关表达式分散到各章去介绍，一方面让读者尽早接触到程序，另一方面也避免了因语言细节而单调无味；第 3～5 章分别介绍程序的 3 种基本结构，体现了最基本的程序设计方法；第 6 章介绍函数，体现了模块化程序设计的需要。前 6 章只涉及 C 语言的基本数据类型，重点放在程序的 3 种基本结构的实现方法和程序设计能力的培养上。第 7～10 章是数组、指针类型和 C 语言的构造数据类型，涉及更复杂数据的表达方法。第 11 章是文件操作，这是程序设计语言的经典内容。这种内容编排符合初学者的认知特点，有利于总体上把握课程内容，帮助读者逐步深入理解和掌握课程知识。各章小结中总结了本章主要的知识点和常见的错误，帮助读者总结归纳课程内容，达到巩固提高的目的。

四是本书有配套的教学参考书、教学课件与相关教学资源。为了方便教学和读者上机操作练习，作者还编写了《计算机程序设计实践教程——C 语言》一书，作为与本书配套的教学参考书。实践教程既与本教材相互配套，又是本教材很好的补充。另外，还有与本书配套的教学课件、各章习题答案、例题源程序等教学资源，可从清华大学出版社网站（http://www.tup.com.cn）下载使用，也可发邮件到 weijj@tup.tsinghua.edu.cn 咨询。

本书第 1～6 章由刘卫国编写，第 7～10 章由童键编写，第 11 章及附录由舒卫真编写，本书由刘卫国、童键担任主编。此外，参与讨论与部分编写工作的还有蔡旭晖、刘胤宏、文碧望、石玉、欧鹏杰、胡勇刚、刘苏洲、孙士闯、周克涛等。清华大学出版社的编辑对本书的策划、出版做了大量工作，在此表示衷心的感谢。

由于编者水平有限，书中难免存在不足之处，恳请广大读者批评指正。

编　者

2014 年 10 月

目 录

第1章 程序设计概述

计算机是在程序(Program)控制下进行自动工作的,它解决任何实际问题都依赖于解决问题的程序,而要编写程序就要熟悉一种程序设计语言,掌握程序设计(Programming)的基本方法,理解计算机分析和解决问题的基本过程和思维规律,为应用计算机进行科学研究与实际应用奠定坚实基础。在众多的程序设计语言中,C语言具有程序简洁、数据类型丰富、表达能力强、使用灵活、实用高效等优点,在当今软件开发中有着广泛的应用。本书以C语言作为实现工具,介绍程序设计的基本思想和方法。

本章介绍程序设计的基本知识、C语言的发展与特点、C程序的基本结构以及C程序的执行步骤。通过本章的学习,使读者对程序设计和C语言有一个概要认识,从而为以后各章的学习打下基础。

1.1 程序设计基础知识

在学习C语言程序设计之前,需要了解一些程序设计的基础知识,包括程序设计的过程、算法的概念、算法的描述方法以及流行的程序设计方法。

1.1.1 程序与程序设计

从一般意义来说,程序是对解决某个实际问题的方法和步骤的描述,而从计算机角度来说,程序是用某种计算机能理解并执行的语言描述的解决问题的方法和步骤。计算机执行程序所描述的方法和步骤,并完成指定的功能。所以,程序就是供计算机执行后能完成特定功能的指令序列。

一个计算机程序主要描述两部分内容:一是描述问题的每个对象和对象之间的关系,二是描述对这些对象进行处理的处理规则。其中关于对象及对象之间的关系是数据结构(Data Structure)的内容,而处理规则是求解的算法(Algorithm)。针对问题所涉及的对象和要完成的处理,设计合理的数据结构可有效地简化算法,数据结构和算法是程序最主要的两个方面。著名的瑞士计算机科学家N. Wirth教授曾提出:

算法+数据结构=程序

程序设计的任务就是设计解决问题的方法和步骤(即设计算法),并将解决问题的方法和步骤用程序设计语言来描述。什么叫程序设计?对于初学者来说,往往把程序设计简单地理解为只是编写一个程序,这是不全面的。程序设计反映了利用计算机解决问题的全过程,包含多方面的内容,而编写程序只是其中的一个方面。使用计算机解决实际问题,通常是先要对问题进行分析并建立数学模型,然后考虑数据的组织方式和算法,并用某一种程序设计语言编写程序,最后调试程序,使之运行后能产生预期的结果。这个过程称为程序设计。具体要经过以

下4个基本步骤。

1. 分析问题，确定数学模型或方法

要用计算机解决实际问题，首先要对待解决的问题进行详细分析，弄清问题的需求，包括需要输入什么数据，要得到什么结果，最后应输出什么。即弄清要计算机“做什么”。然后把实际问题简化，用数学语言来描述它，这称为建立数学模型。建立数学模型后，需选择计算方法，即选择用计算机求解该数学模型的近似方法。不同的数学模型，往往要进行一定的近似处理。对于非数值计算则要考虑数据结构等问题。

2. 设计算法，画出流程图

弄清楚要计算机“做什么”后，就要设计算法，明确要计算机“怎么做”。解决一个问题，可能有多种算法。这时，应该通过分析、比较，挑选一种最优的算法。算法设计后，要用流程图把算法形象地表示出来。

3. 选择编程工具，按算法编写程序

当为解决一个问题确定了算法后，还必须将该算法用程序设计语言编写成程序，这个过程称为编码(Coding)。

4. 调试程序，分析输出结果

编写完成的程序，还必须在计算机上运行，排除程序可能的错误，直到得到正确结果为止。这个过程称为程序调试(Debugging)。即使是经过调试的程序，在使用一段时间后，仍然会被发现尚有错误或不足之处。这就需要对程序做进一步的修改，使之更加完善。

解决实际问题时，应对问题的性质与要求进行深入分析，从而确定求解问题的数学模型或方法，接下来进行算法设计，并画出流程图。有了算法流程图，再来编写程序就容易多了。有些初学者，在没有把所要解决的问题分析清楚之前就急于编写程序，结果编程思路紊乱，很难得到预想的结果。

1.1.2 算法及其描述

计算机是通过执行人们所编写的程序来完成预定的任务。在广义上说，计算机按照程序所描述的算法对某种结构的数据进行加工处理。

算法是对数据运算的描述，而数据结构是指数据的组织存储方式，包括数据的逻辑结构和存储结构。程序设计的实质是对实际问题选择一种好的数据结构，并设计一个好的算法，而好的算法在很大程度上取决于描述实际问题的数据结构。

1. 算法的概念

在日常生活中，人们做任何一件事情，都是按照一定规则、一步一步地进行的，这些解决问题的方法和步骤称为算法。例如，工厂生产一部机器，先把零件按一道道工序进行加工，然后，把各种零件按一定法则组装起来，生产机器的工艺流程就是算法。

计算机解决问题的方法和步骤，就是计算机解题的算法。计算机用于解决数值计算，如科学计算中的数值积分、解线性方程组等的计算方法，就是数值计算的算法；用于解决非数值计算，如用于数据处理的排序、查找等方法，就是非数值计算的算法。要编写解决问题的程序，首先应设计算法，任何一个程序都依赖于特定的算法，有了算法，再来编写程序就会很容易。

下面举两个简单例子，以说明计算机解题的算法。

【例 1-1】 求 $u=\dfrac{x-y}{x+y}$，其中 $x=\begin{cases}a^2+b^2 & a<b\\ a^2-b^2 & a\geqslant b\end{cases}$，$y=\begin{cases}\dfrac{a+b}{a-b} & a<b\\ \dfrac{4}{a+b} & a\geqslant b\end{cases}$。

这一题的算法并不难，可写成：

(1) 从键盘输入 a，b 的值。

(2) 如果 $a<b$，则 $x=a^2+b^2$，$y=\dfrac{a+b}{a-b}$，否则 $x=a^2-b^2$，$y=\dfrac{4}{a+b}$。

(3) 计算 u 的值：$\dfrac{x-y}{x+y}$。

(4) 输出 u 的值。

【例 1-2】 输入 10 个数，要求找出其中最大的数。

设 max 单元用于存放最大数，先将输入的第 1 个数放在 max 中，再将输入的第 2 个数与 max 相比较，较大者放在 max 中，然后将第 3 个数与 max 相比，较大者放在 max 中……一直到比完 9 次为止。

算法要在计算机上实现，还需要把它描述为更适合程序设计的形式，对算法中的量要抽象化、符号化，对算法的实施过程要条理化。上述算法可写成如下形式：

(1) 输入一个数，存放在 max 中。

(2) 用 i 来统计比较的次数，其初值置 1。

(3) 若 $i\leqslant 9$，执行第(4)步，否则执行第(8)步。

(4) 输入一个数，放在 x 中。

(5) 比较 max 和 x 中的数，若 $x>$max，则将 x 的值送给 max，否则，max 值不变。

(6) i 增加 1。

(7) 返回到第(3)步。

(8) 输出 max 中的数，此时 max 中的数就是 10 个数中最大的数。

从上述算法示例可以看出，算法是解决问题的方法和步骤的精确描述。算法并不给出问题的精确解，只是说明怎样才能得到解。每一个算法都是由一系列基本的操作组成的。这些操作包括加、减、乘、除、判断、置数等。所以研究算法的目的就是要研究怎样把问题的求解过程分解成一些基本的操作。

算法设计好之后，要检查其正确性和完整性，再根据它用某种高级语言编写出相应的程序。程序设计的关键就在于设计出一个好的算法。所以，算法是程序设计的核心。

2. 算法的特性

从上面的例子中，可以概括出算法的 5 个特性：

(1) 有穷性。算法中执行的步骤总是有限次数的，不能无止境地执行下去。例如，计算圆周率 π 的值，可用如下公式：

$$\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\cdots$$

这个多项式的项数是无穷的，因此，它是一个计算方法，而不是算法。要计算 π 的值，只能取有限项。例如，计算结果精确到第 5 位，那么，这个计算就是有限次的，因而才能称得上算法。

(2) 确定性。算法中的每一步操作必须具有确切的含义，不能有二义性。

(3) 有效性。算法中的每一步操作必须是可执行的。

(4) 要有数据输入。算法中操作的对象是数据，因此应提供有关数据。但如果算法本身给出了运算对象的初值，也可以没有数据输入。

(5) 要有结果输出。算法的目的是解决一个给定的问题，因此应提供输出结果，否则算法

就没有实际意义。

3. 算法评价标准

在算法设计中，只强调算法特性是不够的。一个算法除了满足5个特性之外，还有一个质量问题。一个问题可能有若干个不同的求解算法，一个算法又可能有若干个不同的程序实现。在不同算法中有好算法，也有差算法。设计高质量算法是设计高质量程序的基本前提。如何评价算法的质量呢？不同时期、不同环境其评价标准可能不同，但一些基本评价标准是相同的。目前，评价算法质量有4个基本标准：

(1) 正确性。一个好算法必须保证运行结果正确。算法正确性，不能主观臆断，必须经过严格验证，一般不能说绝对正确，只能说正确性高低。目前程序正确性很难给出严格的数学证明，程序正确性的证明尚处于研究阶段。要多选用现有的、经过时间考验的算法，或采用科学规范的算法设计方法，是保证算法正确性的有效途径。

(2) 可读性。一个好算法应有良好的可读性，好的可读性有助于保证正确性。科学、规范的程序设计方法(如结构化方法和面向对象方法)可提高算法的可读性。

(3) 通用性。一个好算法要尽可能通用，可适用于一类问题的求解。例如，设计求解一元二次方程 $2x^2+3x+1=0$ 的算法，该算法应设计成求解一元二次方程 $ax^2+bx+c=0$ 的算法。

(4) 高效率。效率包括时间和空间两个方面。一个好的算法应执行速度快、运行时间短、占用内存少。效率和可读性往往是矛盾的，可读性要优先于效率。目前，在计算机速度比较快，内存容量比较大的情况下，高效率已处于次要地位。

4. 算法效率的度量

算法效率的度量分为时间度量和空间度量。

1) 时间度量

算法的执行时间需要依据该算法的程序在计算机上运行时所消耗的时间来度量。它大致等于计算机执行一种简单操作(如赋值、比较等)所需的平均时间与算法中进行简单操作的次数的乘积。因为执行一种简单操作所需的平均时间随计算机而异，它是由所使用计算机的软硬件环境决定的，与算法无关，所以只需讨论影响算法执行时间的另一因素，即算法中进行简单操作的次数。通常把算法中进行简单操作的次数的多少称为算法的时间复杂度，它是一个算法执行时间的相对度量。

一般用问题的规模来表示算法所处理数据的多少。若解决问题的规模为 n，那么算法的时间复杂度就是问题规模 n 的一个函数 $f(n)$，假定时间复杂度记作 $T(n)$，则：

$$T(n) = f(n)$$

例如，求 $s=1+2+3+\cdots+n$，这里问题的规模为 n，求 s 的值需要进行 $n-1$ 次加法，所以算法的时间复杂度 $T(n)=n-1$。

这里算法比较简单，时间复杂度容易计算，当算法比较复杂时，时间复杂度的计算就相对困难。实际上，一般也没必要计算出算法的精确复杂度，只要大致计算出相应的数量级即可，即随着问题规模 n 的增大，算法的执行时间的增长率如何，这个时间增长率称为阶。显然求 s 值算法的执行时间与 n 成正比，所以该算法的时间复杂度是 n 阶的，记为 $T(n)=O(n)$。

算法的时间复杂度采用数量级表示后，将给计算算法的时间复杂度带来很大的方便，这时只需分析影响一个算法的主要部分即可，不必对每一步进行分析。同时对主要部分的分析也可简化，只需分析循环内简单操作的次数即可。例如，下面C语句的时间复杂度为 $O(n^2)$，即是 n 的平方阶的。