高等学校21世纪教材

GAODENG XUEXIAO 21 SHIJI JIAOCAI

C语言程序设计教程

徐士良 编著

COMPUTER

人民邮电出版社
www.pptph.com.cn

高等学校21世纪教材

C语言程序设计教程

徐士良　编著

人民邮电出版社

图书在版编目(CIP)数据

C 语言程序设计教程/徐士良编著.—北京：人民邮电出版社，2001.10
高等学校 21 世纪教材
ISBN 7-115-09204-4

I. C… II. 徐… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2001)第 033260 号

内 容 提 要

本书是作者通过长期教学实践编写成的。全书由浅入深，逐步介绍 C 语言中的基本概念和语法，使读者全面地、系统地理解和掌握用 C 语言进行程序设计的方法。主要内容包括程序设计基本概念、程序的基本组成、选择结构、循环结构、模块设计、数组、指针、结构体与联合体、文件、位运算等。

本书叙述简明扼要，通俗易懂，例题丰富。本书可作为高等院校各专业的学生学习 C 语言程序设计的教材。

高等学校 21 世纪教材
C 语言程序设计教程

◆ 编　　著　徐士良
责任编辑　滑　玉

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件：315@pptph.com.cn
网址　http://www.pptph.com.cn
读者热线：010-67129212　010-67129211(传真)
北京汉魂图文设计有限公司制作
北京朝阳隆昌印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：12
字数：284 千字　　2001 年 10 月第 1 版
印数：1－6 000 册　　2001 年 10 月北京第 1 次印刷
ISBN 7-115-09204-4/TP·2151

定价：17.00 元

丛书前言

当今世界，科学技术突飞猛进，知识经济已见端倪，国际竞争日趋激烈。教育在综合国力的形成中处于基础地位，国力的强弱将越来越取决于劳动者的素质，取决于各类人才的质量和数量，这对于培养和造就我国21世纪的一代新人提出了更加迫切的要求。21世纪初，我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体，是进行教学的基本工具，也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此，高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代，计算机教学内容也随之更新，尤其随着教育部“高等教育面向21世纪教育内容与课程体系改革”计划的实施，对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授，精心策划编写了这套“高等学校21世纪教材”。

为真正实施精品战略，组织编写好这套教材，我们在国内高校做了系统、详细的调查，对教育部制订的教育计划做了认真的研究，还对国内外已出版的教材做了理性的分析，确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中，我们多次组织了由专家及高校一线教师参加的研讨会，对现有比较出色的教材的特点及优点进行了分析，博采众长，力求实现教材权威性与实用性的完美结合。

本套教材有如下特点：

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况，在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。

2. 紧密结合当前技术的新发展，在阐述理论知识的同时侧重实用性。

3. 力求在概念和原理的讲述上严格、准确、精练，理论适中，实例丰富，写作风格上深入浅出，图文并茂，便于学生学习。

4. 为适应当前高校课程种类多、课时数要压缩的教学特点，教材不仅篇幅有很大的压缩，而且均配有电子教案，以满足现代教学新特点的需要，做到易教易学。

5. 所选作者均是国内有丰富教学实践经验的知名专家、教授，所编教材具有较高的权威性。

教育的改革将不会停止，教材也将会不断推陈出新。目前本套教材即将推出，将接受广大教学第一线教师的检验。

由于我们的水平和经验有限，这批教材在编审、出版工作中还存在不少缺点和不足，希望使用本套教材的学校师生和广大读者提出批评和建议，以便改进我们的工作，使教材质量不断提高。

编者的话

程序设计是每个科技工作者使用计算机的基本功。C 语言是目前使用最广泛的一种程序设计语言，具有丰富的数据类型，它所提供的数据结构和控制结构适合于进行结构化程序设计；并且，利用 C 语言可以实现汇编语言的大部分功能，使用灵活，可移植性好。

本书虽然也介绍 C 语言的语法规则，并对一些细节进行了说明，但本书着重强调的是 C 语言的基本概念，通过大量的例题分析和算法实例，使读者理解和掌握利用 C 语言进行程序设计的方法。

全书共分 10 章，每章后面都有充足的习题供读者练习。

第 1 章主要介绍程序设计的基本概念，程序设计的基本过程，以及程序设计的基本方法。通过这一章的学习，使读者对程序设计有一个全面、正确的理解。

第 2 章主要介绍程序的基本组成。这一章是从程序的基本功能出发，结合 C 语言，着重介绍程序中的说明和定义成分，数据的输入与输出功能，以及进行数据处理所涉及到的基本运算。通过这一章的学习，使读者了解和掌握 C 语言中的基本数据类型，基本掌握了实现顺序程序设计的基本运算和相应的基本语句，初步具备了处理简单问题的程序设计能力。

第 3 章主要介绍选择结构的程序设计，利用 C 语言所提供的语句，实现两路分支选择结构和多路分支选择结构。通过这一章的学习，使读者熟练掌握选择结构程序设计的方法。

第 4 章主要介绍循环结构的程序设计，利用 C 语言所提供的语句，实现各种形式的循环结构。通过这一章的学习，使读者熟练掌握循环结构程序设计的方法。

第 5 章主要介绍如何利用 C 语言中的函数实现模块设计。通过这一章的学习，掌握 C 程序中各函数之间数据传递的基本途径以及数据的存储类别，掌握函数调用的正确方法。

第 6 章主要介绍数组在程序设计中的应用。通过这一章的学习，掌握数组在计算机中的存储方式以及用数组作为函数参数时的程序设计方法。

第 7 章主要介绍 C 语言中的指针运算。通过这一章的学习，学会正确使用指针进行程序设计，特别是掌握用指针作为函数参数实现数据的双向传递。

第 8 章主要介绍结构体、联合体以及枚举类型变量的定义和使用。通过这一章的学习，主要掌握用结构体类型实现表格设计的基本方法。

第 9 章主要介绍 C 语言缓冲文件系统中对文件的基本操作。通过这一章的学习，主要掌握对文件的打开与关闭，对文件的读写，以及文件的定位。

第 10 章主要介绍 C 语言中对二进制位的运算以及位段操作。通过这一章的学习，主要掌握各种位运算符的正确使用以及对位段的基本操作。

本书的特点是简明扼要，通俗易懂，例题丰富。与本书配套的还有《C 程序设计题解与实验指导》一书。

由于作者水平有限，书中难免存在错误和不妥之处，恳请读者批评指正。

编　　者

2001 年 4 月

目　录

第1章 绪 论

1.1 程序设计语言的发展

人们要利用计算机解决实际问题，特别是对于数值计算问题，一般总是要编制程序。程序设计语言就是用户用来编写程序的语言，它是人与计算机之间交换信息的工具。

程序设计语言一般分为机器语言、汇编语言和高级语言三大类。

1. 机器语言

机器语言是最底层的计算机语言。用机器语言编写的程序，计算机硬件可以直接识别。在用机器语言编写的程序中，每一条机器指令都是二进制形式的指令代码。在指令代码中一般包括操作码和地址码，其中操作码告诉计算机作何种操作，地址码则指出被操作的对象。对于不同的计算机硬件(主要是 CPU)，其机器语言是不同的，因此，针对一种计算机所编写的机器语言程序不能在另一种计算机上运行。由于机器语言程序是直接针对计算机硬件的，因此它的执行效率比较高，能充分发挥计算机的速度性能。但是，用机器语言编写程序的难度比较大，容易出错，而且程序的直观性比较差，也不容易移植。

2. 汇编语言

为了便于理解与记忆，人们采用能帮助记忆的英文缩写符号(称为指令助记符)来代替机器语言指令代码中的操作码，用地址符号来代替地址码。用指令助记符及地址符号书写的指令称为汇编指令(也称符号指令)，而用汇编指令编写的程序称为汇编语言源程序。汇编语言又称符号语言。

汇编语言与机器语言一般是一一对应的，因此，汇编语言也是与具体使用的计算机有关的。由于汇编语言采用了助记符，因此，它比机器语言直观，容易理解和记忆，用汇编语言编写的程序也比机器语言程序易读、易检查、易修改。但是，计算机不能直接识别用汇编语言编写的程序，必须由一种专门的翻译程序将汇编语言源程序翻译成机器语言程序后，计算机才能识别并执行。这种翻译的过程称为“汇编”，负责翻译的程序称为汇编程序。

3. 高级语言

机器语言和汇编语言都是面向机器的语言，一般称为低级语言。低级语言对机器的依赖性太大，用它们开发的程序通用性很差，普通的计算机用户也很难胜任这一工作。

随着计算机技术的发展以及计算机应用领域的不断扩大，计算机用户的队伍也在不断壮大。为了使广大的计算机用户也能胜任程序的开发工作，从 50 年代中期开始逐步发展了面向问题的程序设计语言，称为高级语言。高级语言与具体的计算机硬件无关，其表达方式接近于被描述的问题，易为人们接受和掌握。用高级语言编写程序要比用低级语言容易得多，并大大简化了程序的编制和调试，使编程效率得到大幅度的提高。高级语言的显著特点是独立于具体的计算机硬件，通用性和可移植性好。

目前，计算机高级语言已有上百种之多，得到广泛应用的有十几种，并且，几乎每一种高级语言都有其最适用的领域。表 1.1 列出了几种最常用的高级语言及其最适用的领域。

表 1.1

语 言 名 称	适 用 范 围
Basic	教学和小型应用程序的开发
Fortran	科学及工程计算程序的开发
Pascal	专业教学和应用程序的开发
C	中、小型系统程序的开发
COBOL	商业与管理应用程序的开发
dBASE	数据库管理程序的开发
FoxBASE	数据库管理程序的开发
C++	面向对象程序的开发
LISP	人工智能程序的开发
PROLOG	人工智能程序的开发
JAVA	面向对象程序的开发

必须指出，用任何一种高级语言编写的程序(称为源程序)都要通过编译程序翻译成机器语言程序(称为目标程序)后计算机才能执行，或者通过解释程序边解释边执行。

1.2 程序设计的基本过程

什么是程序设计？对于初学计算机的人来说，往往简单地把它理解为编制一个程序。其实这是不对的，至少是不全面的。实际上，程序设计包括多方面的内容，而具体编制程序只是其中的一个方面。有人将程序设计描述成如下的一个公式：

程序设计＝算法＋数据结构＋方法＋工具

由此可以看出，在整个程序设计的过程中，要涉及到算法的设计、数据结构的设计、方法的设计和设计工具的选择等诸多方面。从这个概念出发，一般来说，可以将程序设计

的过程分为以下 5 个基本步骤：

① 问题的分析；

② 结构特性的设计；

③ 算法的设计；

④ 流程的描述；

⑤ 调试与运行。

下面分别对这 5 个步骤作简要的介绍。

1. 问题分析

问题分析是进行程序设计的基础。如果在没有把所要解决的问题分析清楚之前就想着手编制程序，是很难得到预想结果的，这只能起到事倍功半的效果。根据所要解决的问题性质与类型，需要分析的内容可能是不同的，但作为最基本的分析内容主要有以下几个方面。

(1) 问题的性质

人们所要解决的问题是各种各样的，而对于不同性质的问题，所用的方法、工具以及输入输出的形式一般也是不同的。通过对问题性质的分析，进一步确定在解决这个问题过程中要做些什么？怎么做？例如，你所要解决的问题是属于数值型还是非数值型的问题；如果是数值型的问题，则要求确定一个合理的精度；不管是数值型问题还是非数值型问题，都需要明确最终的结果是什么。又如，对于一元二次方程求根的问题，需要明确是只求实根还是实根与复根都需要求。

(2) 输入/输出数据

数据处理是计算机应用中最广泛的一个领域。在用计算机解决问题时，一般总要有一些输入数据，计算的结果也要以某种方式进行输出。因此，在进行程序设计的过程中，需要考虑对输入/输出数据的处理。一般来说，对于输入/输出数据主要应考虑以下几个方面。

- 数据的类型是什么？如整型、实型、双精度型、字符型等。
- 在何种设备上进行输入或输出？
- 采用什么样的格式进行数据的输入或输出？

(3) 数学模型或常用的方法

对于数值型问题，一般要考虑数学模型的设计，或者要对常用的一些方法进行分析与比较，从而根据问题的性质选择一种合理的解决方案。对于非数值型问题，通常也需要从众多的方法中选择一种最合理的方法。

例如，为了求一元二次方程 $Ax^2+Bx+C=0$ 的两个实根 x_1 和 x_2，通常有以下 3 种方法：

① 求根公式

$$x_{1,2}=(-B\pm\sqrt{B^2-4AC})/(2A)$$

② 韦达定理

$$x_1+x_2=-B/A$$
$$x_1x_2=C/A$$

③ 迭代法

对于上述 3 种方法，虽然从理论上讲都可以使用，但与实际应用还是有一定的差距。例如，因为计算工具的有效数字位数是有限的，在运算过程中不可避免地会出现误差，因此，利用理论上成立的求根公式计算得到的结果也不一定可靠；韦达定理虽然指出了一元二次方程两个实根之间的关系，但没有指明如何实现的具体步骤；迭代法一般一次只能求一个实根，并且还存在收敛性的问题。因此，在实际解决问题时，必须要对各种方法进行分析比较，不能随便使用某种方法。

2. 结构特性的设计

结构特性设计得好坏，直接影响到程序设计的效率，乃至程序执行的效率。结构特性的设计主要包括控制结构和数据结构的设计。

(1) 控制结构

一个程序的功能不仅取决于所选用的操作，而且还取决于各操作之间的执行顺序，即程序的控制结构。程序的控制结构实际给出了程序的框架，决定了程序中各操作的执行顺序。在程序设计过程中，通常用流程图形象地表示程序的控制结构。常用的流程图有两种：传统流程图与 NS 结构化流程图(简称 NS 图)。

1966 年，Bohm 和 Jacopini 证明了任何复杂的程序都可以用顺序、选择和循环三种基本结构组合而成。其中选择结构中包括两路分支选择结构和多路分支选择结构，其中多路分支选择结构又称为多情况选择结构。循环结构又可以分为当型循环和直到型循环两种。这几种基本控制结构的传统流程图如图 1.1 所示。其中：

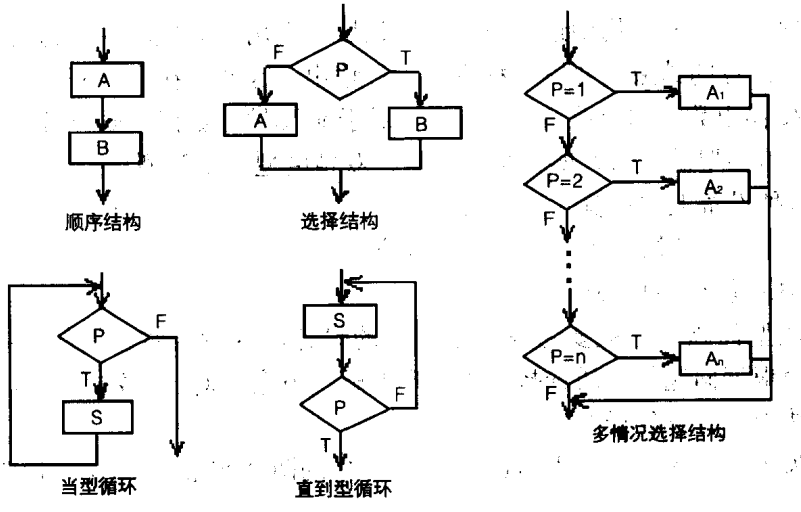

图 1.1　基本控制结构的传统流程图

- 顺序结构反映了若干个模块之间连续执行的顺序；
- 在选择结构中，由某个条件 P 的取值来决定执行两个模块之间的哪一个；
- 在当型循环结构中，只有当某个条件成立时才重复执行特定的模块(称为循环体)；
- 在直到型循环结构中，重复执行一个特定的模块，直到某个条件成立时才退出该模块的重复执行；

• 在多情况选择结构中，根据某控制变量的取值来决定选择多个模块中的哪一个。

传统流程图有以下几个主要缺点。

① 传统流程图本质上不是逐步求精的好工具，它会使程序员过早地考虑程序的控制流程，而不去考虑程序的全局结构。

② 传统流程图不易表示层次结构。

③ 传统流程图不易表示数据结构和模块调用关系等重要信息。

④ 传统流程图中用箭头代表控制流，因此，程序员不受任何约束，可以完全不顾结构程序设计的思想，随意进行转移控制。

在 1.3 节中将介绍结构化流程图(即 NS 图)。

(2) 数据结构

在计算机的各种应用中，数据处理所占的比重将越来越大。在实际应用中，需要处理的数据元素一般有很多，而且，各数据元素之间不仅具有逻辑上的关系，还具有在计算机中实际存储位置上的关系。显然，杂乱无章的数据是不便于处理的，而将大量数据随意地存放在计算机中，实际上也是“自找苦吃”，对处理也是很不利的。

一般来说，在对数据进行处理时，数据的不同组织形式，其处理的效率是不同的。下面的例子说明了数据的不同组织形式对处理效率的影响。

设有一班级学生情况登记表如表 1.2 所示。在表 1.2 中,每个学生的情况是以学号为顺序排列的。显然，要在表 1.2 中查找给定学号的某学生的情况是很方便的。但是，如果要在表 1.2 中查找成绩在 90 分以上的所有学生的情况，则需要从头到尾扫描全表，才能不会漏掉某一个成绩在 90 分以上的学生情况。在这种情况下，成绩在 90 分以下的所有学生也都要被扫描到。由此可以看出，要在表 1.2 中查找成绩在某个分数段中的学生情况，其效率是很低的，尤其是当表很大时问题更为突出。

表 1.2

学　号	姓　名	性　别	年　龄	成　绩
80156	张小明	男	20	86
80157	李小青	女	19	83
80158	赵　凯	男	19	70
80159	李启明	男	21	91
80160	刘　华	女	18	78
80161	曾小波	女	19	90
80162	张　军	男	18	80
80163	王　伟	男	20	65
80164	胡　涛	男	19	95
80165	周　敏	女	20	87
80166	杨雪辉	男	22	89
80167	吕永华	男	18	61
80168	梅　玲	女	17	93
80169	刘　健	男	20	75

为了便于查找成绩在某个分数段中的学生情况，可以将表 1.2 中所登记的学生情况进行重新组织。例如，将成绩在 90 分以上(包括 90 分)、80～89 分、70～79 分、60～69 分之间的学生情况分别登记在 4 个独立的表中，依次如表 1.3～1.6 所示。现在如果要查找成绩在 90 分以上的所有学生情况，则可以直接在表 1.3 中进行查找，避免了对成绩在 90 分以下的学生情况进行扫描，从而提高了查找效率。

表 1.3

学　号	姓　名	性　别	年　龄	成　绩
80159	李启明	男	21	91
80161	曾小波	女	19	90
80164	胡　涛	男	19	95
80168	梅　玲	女	17	93

由这个例子可以看出，在对数据进行处理时，可以根据所需要作的运算不同，而将数据组织成便于运算的形式，以便提高数据处理的效率。

表 1.4

学　号	姓　名	性　别	年　龄	成　绩
80156	张小明	男	20	86
80157	李小青	女	19	83
80162	张　军	男	18	80
80165	周　敏	女	20	87
80166	杨雪辉	男	22	89

表 1.5

学　号	姓　名	性　别	年　龄	成　绩
80158	赵　凯	男	19	70
80160	刘　华	女	18	78
80169	刘　健	男	20	75

表 1.6

学　号	姓　名	性　别	年　龄	成　绩
80163	王　伟	男	20	65
80167	吕永华	男	18	61

3. 算法的设计

前面已经提到，在进行问题分析时，要建立数学模型或对常用的方法进行分析比较，

这实际上就是算法的设计。

所谓算法，是指对解题方案的准确而完整的描述。从程序角度来看，也可以说算法是一个有限条指令的集合，这些指令确定了解决某一特定类型问题的运算序列。

4. 流程的描述

程序设计的过程，实际上就是确定解决问题的详细步骤，而这些步骤通常称为流程。在程序设计过程中，常用的流程描述工具有以下几种。

(1) 自然语言

自然语言是人们在日常生活、工作、学习中通用的语言，一般不需专门的学习和训练就能理解用这种语言所表达的意思。但是，用自然语言描述一个流程时，一般要求直接而简练，尽量减少语言上的修饰。例如，为了描述计算并输出 z=y/x 的流程，可以用自然语言描述如下：

第一步　输入 x 与 y。

第二步　判断 x 是否为 0：

　　若 x=0，则输出错误信息；

　　否则计算 y/x⇒z，且输出 z。

(2) 算法描述语言

为了说明程序的流程，还可以用专门规定的某种语言来描述。这种语言通常称为算法描述语言。算法描述语言一般介于自然语言与程序设计语言之间，它具有自然语言灵活的特点，同时又接近于程序设计语言的描述。但必须指出，用算法描述语言所描述的流程，一般不能直接作为程序来执行，最后还需转换成用某种程序设计语言所描述的程序。算法描述语言与程序设计语言最大的区别就在于，算法描述语言比较自由，不像程序设计语言那样受语法的约束，只要描述得人们能理解就行，而不必考虑计算机处理时所要遵循的规定或其他一些细节。

在程序设计过程中，一般不可能开始就用某种程序设计语言来编制计算机程序，而是先用某种简单、直观、灵活的描述工具来描述处理问题的流程。当方案确定以后，再将这样的流程转换成计算机程序。这种转换往往是机械的，已经不涉及功能的重新设计或控制流程的变化，而只需考虑程序设计语言所规定的语法要求以及一些细节问题。

例如，对于计算并输出 z=y/x 这个例子，其处理的流程可以用一种算法描述语言描述如下：

```
INPUT  x, y
IF (x=0)  THEN
    OUTPUT  "ERROR"
ELSE
    { z=y/x
      OUTPUT  z
    }
```

这样的描述，对于懂一些英语的人，是不难理解的。有时为了使描述更简练，在算法描述语言中还可以使用一些自然语言。例如，上面的描述可以改为：