全国教育科学“十一五”规划课题研究成果

C语言程序设计

Programming in C

李建忠　周　涛　李　征　郭天印　编著

高等教育出版社
HIGHER EDUCATION PRESS

全国教育科学“十一五”规划课题研究成果

C语言程序设计

C Yuyan Chengxu Sheji

李建忠　周　涛　李　征　郭天印　编著

高等教育出版社·北京
HIGHER EDUCATION PRESS　BEIJING

内容提要

本书是一本面向高等学校非计算机专业学生的C语言程序设计教材。本书共分11章。主要内容包括C语言程序概述、C语言中的基本数据与运算、顺序结构实现语句、选择结构实现语句、循环结构实现语句、数组、函数、指针、用户可建立的数据类型、编译预处理与位运算、文件输入输出等。本书贯穿问题驱动的教学理念，强调对学生自主学习和应用能力的培养，内容精练，重点突出。本书还配有《C语言程序设计实训指导书》，可供学生进行课外练习、上机实验和水平测试。

本书可作为高等学校C语言程序设计课程的教材，也可作为C语言程序设计爱好者的自学用书。

图书在版编目(CIP)数据

C语言程序设计/李建忠等编著.--北京:高等教育出版社,2012.2

ISBN 978-7-04-034288-8

Ⅰ.①C… Ⅱ.①李… 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(2011)第279862号

策划编辑 刘 艳　　责任编辑 李善亮　　封面设计 王 洋　　版式设计 杜微言

责任校对 刘春萍　　责任印制 毛斯璐

出版发行 高等教育出版社
社　　址 北京市西城区德外大街4号
邮政编码 100120
印　　刷 北京北苑印刷有限责任公司
开　　本 787mm×1092mm 1/16
印　　张 14.5
字　　数 350千字
购书热线 010-58581118

咨询电话 400-810-0598
网　　址 http://www.hep.edu.cn
　　http://www.hep.com.cn
网上订购 http://www.landraco.com
　　http://www.landraco.com.cn
版　　次 2012年2月第1版
印　　次 2012年2月第1次印刷
定　　价 25.00元

物 料 号 34288-00

前　言

随着信息化进程的不断加快，计算机应用渗透到各行各业，计算机应用技术已成为解决各个专业领域问题的支撑技术。程序设计也成为当代人才知识结构的重要组成部分。因为C语言有着广泛的应用背景和显著的特点，所以成为高等学校程序设计课程的代表语言。

“C语言程序设计”作为以编程方式应用计算机的入门课程，已列为高等学校理工类学生的公共基础课。这门课程教学涉及的专业面广、人数众多、影响深远，其教学质量引起各方面的关注。教材是教学内容体系、教学理念与方法的载体，是提高教学质量的重要保证。近几年，“C语言程序设计”课程的教材建设出现了许多成果，对教学内容体系进行了许多改革，反映了许多新内容、新理念、新方法，但仍有一些值得研究的问题。

在教学背景发生了深刻的变化的情况下，教材应当适应变化。早些年，非计算机专业学生学习程序设计语言，基本上是计算机“盲”的状态。为使学生先接受抽象的程序概念和基本语言内容，教材往往脱离编程语言与计算机内部原理、算法与程序设计等的联系，注重语言规则和形式的详细阐述。近几年，在中学阶段就开设了信息技术课，进入大学的第一学期普遍开设大学计算机基础课，程序设计语言课一般在一年级第二学期后开设，学生都具备一定的计算机基础知识。随着教学背景的变化，在“C语言程序设计”课程的教材建设上虽然也在不断地推陈出新，但在基本内容体系上，尤其在语言内容的认知上，现在的一些教材和前几年的教材没有太大的差别。主要表现在以下几个方面。

① 虽然内容有所更新，但属于添加，使得教材篇幅越来越大。例如，一些教材安排专门的篇章讲解算法和程序设计方法。这些内容本来就十分广泛，涉及较深的基础知识。这门课毕竟是程序设计的入门课，同时受课时的限制，学生对算法、程序设计方法、数据结构等问题的理解达不到预想效果，反而给教师使用教材带来困惑。应该在提高对C语言内容认知的基础上，再融入这些内容。例如，C语言中的数据类型就是用来表示一种数据结构，结构语句就是表示一种算法，函数就是实现模块化程序设计方法等。

② 大多数教材仍沿用按编程语言本身的内容来组织结构体系。本质上，编程语言的应用是程序设计，应该在程序结构的框架下来学习语言。但目前学生反映的情况往往是“学习C语言时，不知道要学些什么，学了能干什么，课程结束时才知道要学什么，能干什么，懊悔领悟晚矣”。这就说明：没有从认知、应用层面上讲解知识，这样就导致学生在学习语句时与程序结构联系不起来，读、写程序时与算法联系不起来，接触编程问题时也就感到无从下手，总觉得内容抽象、多而散，抓不住主线。

③ 脱离原理内涵讲语句，对学生来说，存在一听就懂，听后就忘的问题。编程语言中的语句是在计算机中执行并实现一个功能的，摆脱计算机的执行原理反而会更抽象，不容易理解。例如，“变量”的实质就是标志存储器中存放一个数据的单元，存储器单元中的数据会随着程序的处理不断更新，所以称其为变量。摆脱这一实质意义，学生就会与数学中的变量弄混，很难正确理解。又如，“指针”就是在C语言中对存储器的一种访问机制，不从对存储器访问的原理过程

出发讲解指针,学生就很难抓住指针的实质意义。

④ 一些教材贯穿了案例教学的理念,在一些章节前面往往摆出一个案例程序。学生在不具备必要基础之前,根本看不懂程序,更谈不上理解程序的功能与思路,反而有望而生畏的感觉。如果教师对教材处理不当,也会带来本末倒置的效果。

结合对教学实践中的问题探索,本书力图体现以下特色:

① 注重思路方法的传递。从书的引言到每一章节前都有对相应内容的体系、内在规律、学习目标的概述,旨在让学生抓住思路线条,先知道为什么学、学什么、怎样学,作为学习方法的指导。例如,在本书的引言中概述了:(a) 为什么要学习程序设计?(b) 程序设计涉及哪些知识和技术(程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具)?由此对算法、数据结构、程序设计方法作了简要介绍。(c) 程序设计语言是人与计算机进行交互的语言,抓住语言属性,从熟悉的自然语言体系出发,揭示 C 语言程序设计的内容体系,同时又从计算机工作的机器属性说明其特点,这样可以给学生提供一种学习 C 语言的思路方法。不仅使学生知道学习 C 语言程序设计课程的目的是学会编写程序,而且知道怎样一步一步地学习编写程序。

② 改变对 C 语言内容的认知方式。从语言形式入手讲解内涵规律,从程序的功能结构入手组织篇章结构,把算法落实在语言结构和程序设计过程中。从程序的功能上看,一个程序不外乎是对用户给定数据,按照一定的规则进行加工处理,最后将处理结果提供给用户;从程序结构上看,程序设计语言包括顺序、分支、循环处理结构语句;从模块化程序设计方法看,包括模块(函数)的定义、调用、数据传递的内容;从数据结构看,数据包括简单数据、复杂数据。本书综合地考虑了程序的各种功能结构关系,按照由简单到复杂,由浅入深的认知规律组织了篇章内容体系。几乎每一章标题都有一个主标题和一个副标题,如“第 6 章数组——批量数据的表示与处理”,主标题表示语言内容,副标题表示程序功能、结构、方法方面的内容,这样便于把语言基本内容与程序功能、结构、方法有机地联系起来,每学一部分内容,编程能力就得到一定程度的提高。

③ 把案例教学的理念变为问题驱动的教学理念。根据内容先恰当地提出问题,然后围绕要解决的问题讲解内容,最后得到最终的程序。这样更符合认知规律,更能激发学生的学习兴趣。

④ 注重应用与实践。在每一个知识单元后都有相应问题的应用程序设计实例和习题。

⑤ 注重在提升知识认知的基础上,挖掘内涵与规律,突出重点与难点,缩略细节,压缩篇幅,减轻教师使用教材的难度,也便于学生自学。

本书第 1 章、第 4 ~ 8 章由李建忠编写,第 2 ~ 3 章由周涛编写,第 9 ~ 10 章由李征编写;第 11 章和附录由郭天印编写,全书由李建忠统稿。

本书的编写工作得到了陕西理工学院领导和教务处的高度重视和大力支持,在此表示衷心的感谢。

编者力图在教材中反映一些对“C 语言程序设计”课程教学改革的思路与方法,编写出具有一定特色的教材,但受水平限制,或许只能起到抛砖引玉的作用,书中难免存在不妥或错误之处,敬请批评指正。

编 者

2011 年 10 月

目　录

第1章　C语言程序设计概述

在学习C语言程序设计的具体内容之前，对程序设计语言、程序设计的基本方法、数据结构与算法、程序设计语言的内容体系、C语言的基本特点、C语言程序结构有一个初步了解，有利于明确学习目标与内容、树立正确的思维与方法。本章正是从这个角度出发，对上述内容作简要概述。

1.1　程序设计语言

从计算机外部看它的工作，似乎计算机具有人的智能，能实现人的各种意图。其实，计算机是程序控制的高度自动化的信息处理工具。要让计算机来实现人的意图，必须将意图编写成程序，输入并存储到计算机的存储器中，只有当计算机执行程序，才能按人的意图进行规定的操作。也就是说，没有程序，计算机就表现不出任何功能。

程序是解决某一问题的方法步骤在计算机中的表达或描述。这种表达或描述必须采用人与计算机能进行交互的语言，即程序设计语言。

随着计算机应用技术的发展，程序设计语言也经历了不同阶段的发展，出现了多种语言。从人与计算机的交互性能角度，程序设计语言可分为低级语言和高级语言阶段。低级语言包括机器语言、汇编语言，高级语言又分为面向过程的高级语言和面向对象的高级语言。

机器语言是计算机直接能够识别的语言。计算机设计时，为它所能进行的每一个操作设计一条二进制编码指令。采用这种编码指令与计算机进行交互，就称为机器语言。在计算机应用初期，人们用机器语言编写程序。但是机器语言是冗长的二进制代码，难理解、难记忆、难编程，只有少数计算机专业人员才会使用。随着计算机应用技术的发展，计算机语言一直朝着人性化的方向发展，先后出现了汇编语言和不同种类的高级语言。

汇编语言是用一组便于人们记忆的助记符号来表示机器语言指令。汇编语言中一条指令与一条机器指令相对应。机器语言和汇编语言都是面向具体计算机的语言，每一种类的计算机都有自己的机器语言和汇编语言。由于它们依赖具体的计算机，所以被称之为低级语言。

高级语言不依赖具体的计算机，是在各种计算机上都通用的程序设计语言。高级语言接近人们习惯使用的自然语言和数学语言，易于学习和使用。20世纪50年代出现了高级语言。人们认为，高级语言的出现是计算机发展史上一次惊人的成就，使非计算机专业人员能方便地编写程序，用计算机来解决各专业领域的问题。随着计算机应用技术的发展，高级语言又分成了两类：一类是面向过程的语言，另一类是面向对象的语言。

(1) 面向过程的语言

所谓过程，其实质是某个输入集合到某个输出集合的一个映射，可以用某种描述形式来说明这种映射的细节。从解决问题的角度讲，可以把解决问题方式抽象为较大的过程，过程中还可能分解为较小的过程，一直到具体的操作步骤。这些过程都可以用某种高级语言按照一

定的顺序和规则进行描述。这种类型的程序设计语言称之为面向过程的高级语言。也就是说,用面向过程的语言编程只需根据解决问题的过程,把过程的每一步骤按顺序用语言描述出来,即基于算法的编程语言。面向过程的编程语言有 BASIC、FORTRAN、COBOL、Pascal、C、Ada、LISP 等。

(2) 面向对象的语言

面向对象的语言是用客观世界中描述事物的方法来描述一个程序要描述的事物,使程序设计更接近现实世界的思维方式,更简捷高效。面向对象的程序设计是以处理的数据为对象,根据对象的属性与操作进行封装并定义接口来组织程序。常用的面向对象的程序设计语言有 Visual C++ 、Visual Basic、Delphi、Java 等。

计算机编程语言只是提供了一种人与计算机进行交互的方法或工具。计算机唯一能识别的语言只有机器语言,所有非机器语言编写的程序都要经过转换为机器语言的环节,才能在计算机中执行。这个转换环节一般称为编译。编译都由系统软件来完成。所以,为了方便用户使用,任何高级语言都有一个实现编译等操作功能的语言处理软件。C 语言的语言处理软件有多种,如 Turbo C、Visual C++ 6.0 等。

由非机器语言编写的程序一般称为源程序,能被计算机直接执行的程序称为可执行程序。把一个源程序变换成一个可执行程序,一般要经过图 1.1 所示的处理。

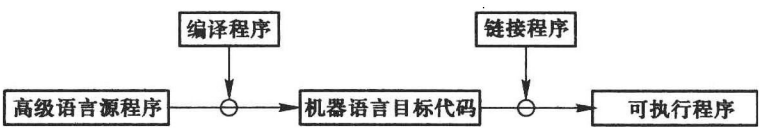

图 1.1 源程序转换处理步骤

高级语言的种类很多,不同的语言有不同的特点与使用场合,但从原理上看,各种语言都包含一些相同功能和结构。对初学程序设计语言者来说,只要选择有代表性的语言,掌握了程序设计语言规律和内在功能结构,就很容易学习和应用其他语言。现在一般都把 C 语言作为程序设计的入门语言。

1.2 程序设计的基本方法

程序设计涉及多方面的知识和技术。可以将程序设计所包含的内容表示为:

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具

程序设计是四个方面知识的综合运用与贯通。事实上,这四个方面中的每一个内容都属于有关专门课程的范畴。本书只针对 C 语言来学习编程语言工具,但对算法、数据结构、程序方法应有一个初步了解,便于理解和掌握 C 语言程序设计。

1.2.1 数据结构与算法

数据结构与算法具有密切的联系。因为数据结构是加工对象,算法是对数据对象加工处理的方法。不同的数据结构可能需要采取不同的算法,不同的算法可以采用不同的方式对数据进

行加工处理。

(1) 数据结构

数据结构指的是数据的组织形式,如字符数组、方程组系数矩阵、人员的基本信息、反映部门组织机构关系的树形图、反映网络结构及信息的网状图等都具有不同的数据结构。

一般情况下,数据元素之间不是孤立的,常常存在某些逻辑上的联系,这些联系与存储无关,独立于计算机。我们把客观事物本身的逻辑关系称为逻辑结构。

数据要能被计算机处理,就必须存储在计算机内,把数据元素及其关系在计算机内的表示称之为数据的存储结构,它是数据的逻辑结构在计算机存储器中的映射。

组织和存储数据的目的是能被计算机处理,即对数据施加各种运算。因此,可以在逻辑结构上定义运算集合,而在存储结构上实现这些运算。

在程序设计语言中,一般是以数据类型来表示一种数据结构,定义一种数据类型既表示了逻辑结构的组织,同时也就定义了存储结构。例如,C 语言中整型、实型、字符型、数组、结构体与共用体等都表示相应的数据结构。

(2) 算法

所谓算法是解决一个问题所采取的方法和步骤。程序设计中的算法是把解决问题的每一步骤具体化为计算机的操作,即算法是解决计算机在什么情况下应该“做什么”和“怎样做”的问题。

软件行业把软件开发分为设计和编码两个不同的阶段。所谓设计就是算法的设计,编码是对算法采用编程语言来表示。算法的设计就是算法的分析与表示。算法设计没有固定的模式。对于同一个问题可有多种算法,不同的设计者可以设计出不同的算法。下面通过一个举例使读者理解算法及算法设计。

例 1.1 把一个班 30 个学生中一门课考试成绩不及格学生的学号和成绩打印出来。

假设用 c 表示学生的学号,c_i 代表第 i 个学生的学号,用 s 表示学生的成绩,s_i 代表第 i 个学生的成绩,问题的算法可表示如下。

步骤 1:1⇒i(相当于设立了一个指针,先指向第一个学生)

步骤 2:如果 $s_i<60$,则打印出学号 c_i 和成绩 s_i,否则不打印

步骤 3:i+1⇒i(相当于修改指针,使其指向下一个学生)

步骤 4:如果 i≤30,返回步骤 2,继续执行,否则算法结束

算法要采用一定的形式表示出来,其目的是便于用一种计算机编程语言把算法变换成能在计算机上实现的程序。算法的描述有多种方式,如图解方式、语言方式等。其中,流程图是最容易理解的一种最基本的算法描述方法。下面对流程图进行简单介绍。

流程图采用一些规定的图形符号来表示相应的处理流程。常用的图形元素如图 1.2 所示。

前面算法举例,用流程图的表示如图 1.3 所示。

流程图的另一种表示形式是 N-S 图。N-S 图是去掉流程线、把算法写在一个框内的流程图。几种基本程序结构的 N-S 使用的符号如图 1.4 所示。图 1.3 所示算法的 N-S 图表示如图 1.5 所示。

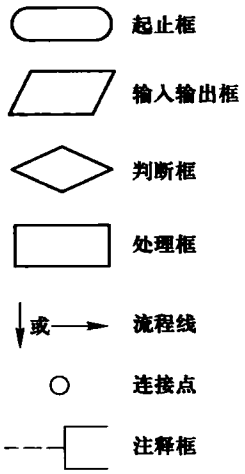

图 1.2　流程图使用基本图形

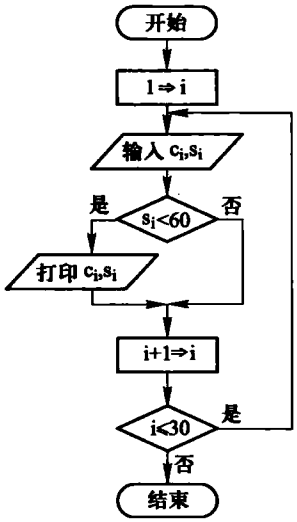

图 1.3　算法流程图

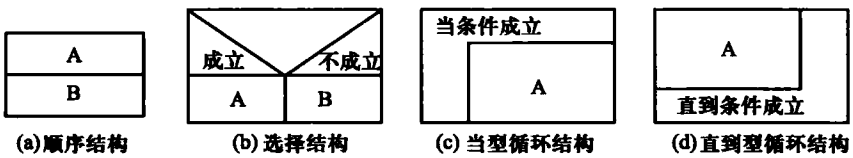

图 1.4　N－S 图基本结构图形

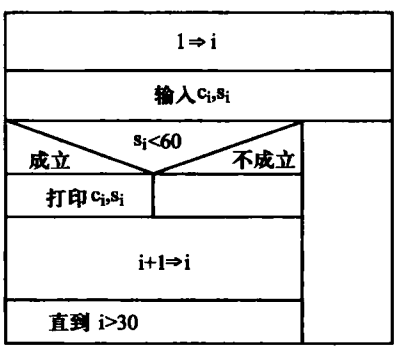

图 1.5　N－S 图示例

1.2.2　程序设计方法

程序设计方法涉及非常丰富的内容，但模块化程序设计思想始终贯穿在所有程序设计过程中。

模块化程序设计方法的基本思想就是抽象和分解。

抽象是人类认识世界的基本法则之一，在现实世界中一些事物、状态或过程之间总是存在着某些相似的方面，把这些方面集中和概括起来，暂时忽略它们之间的差异，或者说抽出事物的本质特性而暂时不考虑其细节，这就是抽象。

抽象包含了系统的观点和分层的观点，即可以把程序设计问题看成是一个系统，这个系统可用高级的抽象概念来理解和构造，这些高级的抽象概念又可用较低级的抽象概念来理解和构造，如此进行下去，直到最底层的模块可以表示成某种程序设计语言的语句为止。

实际上，对系统的每一次抽象都是向更具体的方面推进的过程，也是进一步分解的过程。因此，在抽象的过程中，同时伴随着分解。当遇到一个较大的、较复杂的问题时，不要急于编写程序代码，而应该首先把问题自顶向下、逐步分解、细化成一个个程序模块，然后再对每一个模块进一步细化。处于不同层次的模块应该只考虑本模块内部的问题而不必考虑其他模块内部的问题。顶层模块控制了系统的主要功能并影响全局，底层模块则完成对数据的具体处理。

信息隐蔽和局部化是模块化程序设计方法的另一基本思想。这一思想是指：在模块化的过程中，使一个模块内包含的信息不能被不需要这些信息的模块访问。也就是说，模块的划分应该遵循独立性原则，即模块之间互相依赖的程度要小，而模块内部各元素彼此结合的程度要大。

程序设计的重要方法是结构化程序设计。结构化程序设计是以模块设计为中心，任何简单或复杂的算法都可以由顺序结构、选择结构和循环结构这三种基本结构组合而成。所以，这三种结构就被称为程序设计的三种基本结构。

1.3 程序设计语言的内容体系

对初学程序设计语言者来说，在不知道语言体系的情况下，往往不知道程序设计语言包含哪些内容，内容之间有什么作用关系。在学习基本内容时，感觉十分抽象，本来是一些简单的规则，都会感到深奥莫测，学习效果不理想。先对程序设计语言体系有一个初步了解，能明确学习目标和内容，对学习有很大的帮助。

程序设计语言是人与计算机进行交互的语言。它既具有自然语言的要素和规律，又有计算机处理所要求的一些特点。

为了从自然语言体系来联想程序设计语言体系，用图 1.6 描述了自然语言和程序设计语言的联系。从图 1.6 可以看出，程序设计语言与自然语言具有可类比的体系内容。但程序语言与自然语言又有很大差别。自然语言中，在字词及符号的基础上，依据语法规则造句。程序设计语

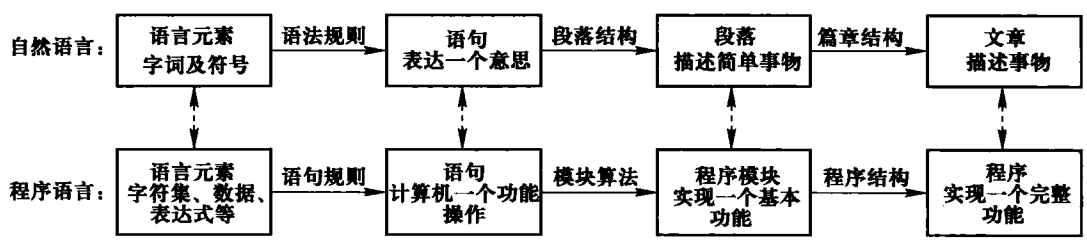

图 1.6 程序设计语言体系

言中,每一语句表示计算机内部的一个功能操作,都有严格的格式规定,必须按照规定书写语句,否则计算机系统就不能识别。程序是依据算法由相应的功能语句构成的,要求计算机实现一个特定的功能。学习了简单数据对象和简单算法的结构语句就可以编写简单程序,进一步学习复杂数据对象、复杂算法结构语句及结构就可以编写复杂程序。

1.4 C语言的特点

C语言是在20世纪70年代初,由贝尔实验室D. M. Ritchie在总结之前高级语言优缺点的基础上,并针对UNIX操作系统而设计出的高级编程语言。随着UNIX操作系统的广泛应用,C语言很快风靡世界,成为最流行的程序设计语言。在C语言的发展初期,曾出现了多种版本。为了达成一致,1983年美国国家标准学会(ANSI)制定了C语言标准,称为ANSI C。随着计算机应用技术的发展,C语言也经历了不断更新与发展的过程。1990年,国际标准化组织(International Standard Organization,ISO)接受C89作为国际标准。此后又经历了几次修订与增补,最近的标准版本是C99。本书参照C99标准来讲述C语言程序设计。

C语言具有一些卓越的特点。C语言内部具有许多功能结构和应用特点,这些特点只有在学习C语言的过程中才能领悟,下面仅对C语言的一般特点作简要说明。

① 语言简洁,使用灵活,易于学习和使用。C语句接近于自然语言和数学语言,功能与语义紧密联系,容易理解和应用。

② 数据类型丰富。C语言包含丰富的数据类型,涵盖了实际问题所面临的各种数据结构,几乎不需要用户自己定义数据结构,给程序设计带来很大便利。

③ 数据处理功能强。C语言包含数值运算符、字符运算符、逻辑运算符和字/位运算符,与丰富的数据类型结合,构成灵活多样的表达式,可以实现其他高级语言中难以实现的运算。

④ 具有低级语言对计算机的一些硬件资源进行控制的功能。C语言不但具有其他高级语言的功能和应用方式,而且还兼有低级语言的许多功能,如对存储器或I/O端口的地址访问、位操作功能等,不仅是通用的程序设计语言,又广泛地应用于系统软件开发和实时控制应用系统开发。

⑤ 既支持结构化程序设计,又支持面向对象程序设计。随着程序设计方法和技术的发展,在基本C语言的基础上,又产生了C语言的超集C++,二者具有良好的兼容性和继承性,使得基本C语言支持结构化程序设计,C++支持面向对象程序设计。学习基本C语言是学习C++的基础。C语言程序可以在C++编译系统运行,本书中所有程序都是在Visual C++ 6.0环境中成功运行的。

⑥ 生成的目标代码效率高。只有汇编指令与机器码指令一一对应,汇编语言源程序生成的目标代码几乎不包含冗余码。一般高级语言源程序生成目标代码都包含一定量的冗余代码,C语言源程序生成的目标代码包含的冗余代码最少,其目标代码的效率仅比汇编语言低10%~20%。

⑦ 可移植性好。C语言程序几乎可在各种类型计算机和操作系统上运行。

1.5 基本 C 语言程序结构

在学习 C 语言具体内容之前,对基本程序结构有一个初步了解,一方面能具体认识 C 语言的一些特点,另一方面能认识一个 C 语言程序的基本组成,对后面的学习是很有好处的。下面通过一个简单 C 语言程序来说明其构成特点。

例 1.2 求从键盘上输入两个整数之和,并将结果输出。

```
#include < stdio. h >                    //包含标准输入输出函数信息定义的头文件
int main( )                             //主函数
{
    int a,b,s;                          //定义 3 个数据变量
    scanf( " % d,% d" ,&a,&b) ;         //输入两个求和变量的数值
    s = a + b;                          //实现两个变量求和
    printf( " sum = % d\n" ,s) ;        //求和结果输出
}
```

例 1.2 运行结果:

```
199,201
sum=400
```

针对上面的例子程序,可以从功能、结构上了解一个 C 语言程序的构成特点。

1.5.1 从结构上看 C 语言程序的构成

从结构上看一个 C 语言程序有以下结构特点。

① C 语言程序由函数构成。函数是 C 语言程序的基本单位,是一个实现特定功能的程序模块,相当于其他语言中的子程序。一个 C 语言程序必须有一个主函数,并用专用名字 main,程序从 main( )函数的第一条可执行语句开始执行。一个 C 语言程序可以包含若干个函数(系统提供的库函数和用户定义的函数),由函数调用关系形成完整的功能体系。从这个角度来说,C 语言程序的设计,就是一个个函数及其调用关系的设计。在例 1.2 程序中,因功能简单,所以只有一个主函数。

② 一个函数由函数首部和函数体两部分构成。因为 C 语言程序中函数是一个功能相对独立的模块,所以在函数首部要定义本函数被引用的名称及其模块之间需传递的参数;函数体是功能具体实现部分,函数体被一对大括号界定。具体的函数定义的内容将在后续章节中学习。

③ 函数体中每一个语句用“;”作为结束符。

④ 若在一个 C 语言程序中有标准库函数调用,在程序首部应用预处理命令#include 把有关信息包含在本程序中。C 语言编译系统为各类标准库函数提供了原型定义,分别包含在扩展名为 . h 的头文件中,使用哪一类库函数,应用#include 把相应的头文件包含进来。

⑤ 在 C 语言程序的任一语句行可以用“//”做注释。注释只是为了提高程序的可读性,对程序的功能及处理没有丝毫影响。注释内容可以采用程序员习惯的语言字符。

1.5.2 从功能上看C语言程序的构成

从功能讲,一个程序就是实现对特定数据存储、处理,将处理结果输送给用户的过程,所以应包含提供原始数据、数据存储、数据处理、数据输出这几个环节的操作语句。

① int a,b,s;是为存储数据定义了3个变量。在C语言程序中,一个变量代表存放一个数据的单元。

② scanf("%d,%d",&a,&b);是提供原始数据的语句,即从键盘上输入两个运算数。

③ s=a+b;是对数据加工处理的语句,即实现两数相加的运算。

④ printf("sum=%d\n",s);是将处理结果输送给用户的语句。没有这个语句,就得不到程序的运行结果。

1.5.3 C语言程序设计规范

利用C语言编写程序,必须遵守C语言系统的基本规定,如基本字符集、各种关键字、符号命名规则、语句格式、函数定义规则和程序结构等。为了提高程序的可读性和软件开发的效率,在C语言系统的基本规定的基础上,还有不同的编程规范。附录E列出了一种参考编程规范。在初学C语言程序设计时,可以暂不关心C语言系统基本规定之外的编程规范,但建立编程规范意识和习惯,对将来开发实际应用系统是很有好处的。

习 题

1.1 计算机通过什么方式来实现人的意图?

1.2 什么是程序?程序设计包含哪些内容?

1.3 编程语言有哪些种类?各有何特点?

1.4 何谓数据结构?何谓算法?它们与程序设计有何关系?

1.5 表达算法的方式有哪几种?

1.6 请画出从a、b、c三数中求出最大数的流程图。

1.7 简述模块化程序设计方法的主要思想。

1.8 简述程序设计语言与自然语言有哪些异同?

1.9 简述C语言的特点。

1.10 一个简单的C语言程序从结构上应包含哪些部分?从功能上应包含哪些部分?