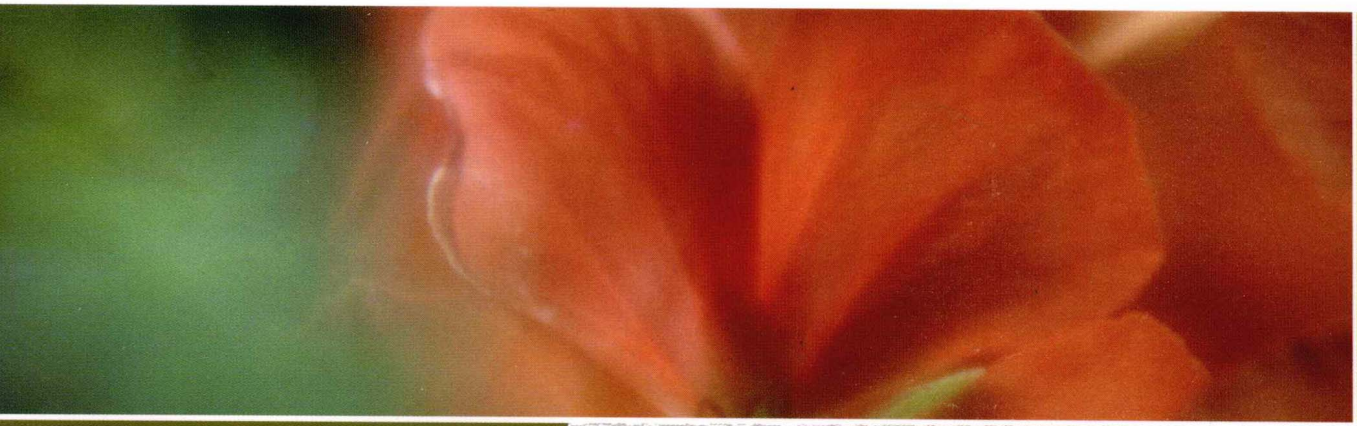
21世纪高等学校规划教材

C语言程序设计

韩增红 王冬梅 主编



21st Century University Planned Textbooks

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校规划教材

C语言程序设计

韩增红 王冬梅 主编
佟继红 李明 肖丽君 段立平 毕国忠 张泽梁 胡智鹏 编著

21st Century University
Planned Textbooks

人民邮电出版社
北京

图书在版编目（CIP）数据

C语言程序设计／韩增红，王冬梅主编. —北京：人民邮电出版社，2009.10
21世纪高等学校规划教材
ISBN 978-7-115-21357-0

Ⅰ. C… Ⅱ. ①韩…②王… 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（2009）第158936号

内 容 提 要

本书以C语言程序设计的基本原理为出发点，以应用为主线，内容讲解由浅入深、循序渐进、重点突出。本书的特点是概念准确、内容合理、案例丰富、实用性强。

全书共分12章，内容包括：概述、数据类型、运算符和表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、函数、构造数据类型、指针、文件、音乐与图形设计及综合应用。每章后都附有适量的习题，读者可通过习题巩固已学的知识。书中全部程序均上机调试通过。

本书可作为本科、专科及各类成人教育的C语言程序设计教学用书，也可作为计算机培训和计算机等级考试的教材，还可作为广大程序开发人员和计算机爱好者学习C语言程序设计的参考书。为配合本书的学习，本书配有《C语言程序设计上机指导与习题》辅导用书，供读者参考。

21世纪高等学校规划教材

C语言程序设计

◆ 主　　编　韩增红　王冬梅
　编　　著　佟继红　李　明　肖丽君　段立平　毕国忠
　　张泽梁　胡智鹏
　责任编辑　滑　玉
　执行编辑　武恩玉

◆ 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@ptpress.com.cn
　网址　http://www.ptpress.com.cn
　北京鑫正大印刷有限公司印刷

◆ 开本：787×1092　1/16
　印张：19.75
　字数：517千字　　2009年10月第1版
　印数：1－3 000册　　2009年10月北京第1次印刷

ISBN 978-7-115-21357-0

定价：33.00元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

C 语言是一种得到广泛重视并普遍应用的计算机程序设计语言。它因其功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有高级语言的优点，又具有低级语言等诸多特点，而成为当今软件开发领域中广泛使用的一种语言。C 语言既可用来编写系统软件，也可用来编写应用软件，是国际公认的最重要的几种通用程序设计语言之一，也是国内外大学介绍计算机程序设计方法的首选语言。

本书以 C 语言程序设计的基本原理为出发点，以程序设计为主线，以实际应用为目标，内容讲解由浅入深、循序渐进、重点突出，内容体例安排合理、案例丰富、实用性强。

全书共分 12 章。第 1 章概述，简单介绍了程序设计的基础知识、C 语言的特点、程序结构和上机步骤；第 2 章数据类型、运算符和表达式，介绍了数据在计算机内的表示、C 语言的数据类型、常量、变量及运算符和表达式；第 3 章至第 5 章详细介绍了 C 语言的结构化程序设计方法，包括顺序结构、选择结构和循环结构程序设计；第 6 章数组，介绍了各类数组的定义和使用方法；第 7 章函数，详细介绍了 C 语言程序的结构、函数的定义及使用，并简单介绍了程序编译预处理；第 8 章构造数据类型，讨论了结构体、共用体、枚举类型的定义及使用；第 9 章指针，深入浅出地介绍了指针的概念和应用；第 10 章文件，介绍了文件的概念和对文件的各种操作；第 11 章音乐与图形设计，介绍了实用的音乐设计和图形设计方法；第 12 章综合应用，从结构化程序设计方法学角度出发，阐述了 C 语言开发应用程序的一般步骤和方法。各章后都附有适量的习题，读者可通过习题巩固已学的知识。书中全部实例和习题均已上机调试通过。

本书既可作为高等院校本专科学生的教材，也可作为计算机等级考试及其他计算机应用人员学习高级语言程序设计的参考书。为配合本书的学习，本书配有《C 语言程序设计上机指导与习题》辅导教材，供学习者参考。

本书由韩增红、王冬梅主编，参加编写的还有佟继红、李明、肖丽君、段立平、毕国忠、张泽梁、胡智鹏等。韩增红编写第 1 章、第 7 章，王冬梅编写第 2 章，段立平编写第 3 章，毕国忠编写第 4 章，胡智鹏编写第 5 章，肖丽君编写第 6 章、第 10 章，佟继红编写第 8 章、第 11 章，李明编写第 9 章，张泽梁编写第 12 章，孙淑霞、许盟参加了本书的部分内容编写及程序调试。

由于编者水平有限，书中难免有不足之处，恳请读者提出宝贵意见和建议。

编　者

2008 年 8 月

目 录

第1章 概述

计算机语言是人与计算机之间交流信息的工具，由计算机能够识别的语句组成，它使用一整套带有严格规定的符号体系来描述计算机语言的词法、语法、语义、语用。词法负责从构成源程序的字符串中识别出一个个具有独立意义的最小语法单位（单词）；语法涉及语言的构成规律，确定程序的结构形式；语义说明语句代表的含义及该语句的执行过程；语用指出语句的实际用途。

C 语言是一种通用的程序设计语言，它具有丰富的运算符和表达式，以及先进的控制结构和数据结构。C 语言既具有高级语言简单易学和可移植性好的特点，又具有汇编语言生成代码质量高的优点。因此，C 语言具有较强的生命力和广泛的应用前景。

本章从程序设计基础知识入手，对 C 语言作一概括性介绍，让读者了解一个 C 语言程序的基本框架和它的书写格式。使读者能够学会编写简单的 C 程序，并能够进行编辑、编译、连接、调试运行等上机操作。

1.1 程序设计基础

在介绍 C 语言程序设计之前，我们先来了解一些有关程序设计的基础知识。

1.1.1 程序与程序设计语言

1. 程序

所谓程序，就是一系列遵循一定规则和思想并能正确完成指定工作的代码（也称为指令序列）。简单地说，程序主要用于描述完成某项功能所涉及的对象和动作规则。通常，一个计算机程序主要描述两部分的内容，其一是描述问题的每个对象及它们之间的关系，即数据结构的内容；其二是描述对这些对象进行处理的动作、这些动作的先后顺序以及它们所作用的对象，要遵守一定的规则，即求解某个问题的算法。

因此，对程序的描述，也可以用经典的公式来表示：

程序=数据结构+算法

一个设计合理的数据结构往往可以简化算法，而且一个好的程序应该具有可靠性、易读性、可维护性等良好特点。

计算机程序有以下共同的性质。

（1）目的性：程序有明确的目的，运行时能完成赋予它的功能。

（2）分步性：程序为完成其复杂的功能，由一系列计算机可执行的步骤组成。

（3）有序性：程序的执行步骤是有序的，不可随意改变程序步骤的执行顺序。

（4）有限性：程序是有限的指令序列，程序所包含的步骤是有限的。

（5）操作性：有意义的程序总是对某些对象进行操作，使其改变状态，完成其功能。

2. 程序设计语言

我们平时在使用计算机工作时，计算机所做的工作实际上是由人们事先编好的程序来控制的，编写程序的工具就是程序设计语言。

程序设计语言（Programming Language）是一组用来定义计算机程序的语法规则。它是一种被标准化的交流技巧，用来向计算机发出指令。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的行动。

程序设计语言有许多种，按照程序设计语言发展的过程，大概分为3类：机器语言、汇编语言和高级语言。

（1）机器语言

机器语言是由二进制代码0和1按一定规则组成的、能被机器直接理解和执行的指令集合。它的突出优点是具有最快的运行速度和最少的存储开销；它的缺点是机器指令不容易记忆，可读性差，编程工作量大，容易出错，通用性差，且随着不同的计算机系统而改变，现在已经没有人用机器语言直接编程了。

下面这3行就是某计算机的机器指令，功能是计算A=15+10的值：

10110000 00001111

00101100 00001010

11110100

首先，把15放入累加器A中，然后让10与累加器A中的值相加，结果仍放入A中，最后结束，停机。

（2）汇编语言

为了克服机器语言的缺点，人们用英文助记符描述机器指令，如用ADD表示加、SUB表示减等，这种采用指令助记符表示的语言就是汇编语言，又称符号语言。例如，上述A=15+10的计算可用下面的汇编语言程序实现。

```
MOV  A,15          ：把15放入累加器A中
ADD  A,10          ：10与累加器A中的值相加，结果仍放入A中
HLT                ：结束，停机
```

这3条汇编指令与前面3条机器指令的作用是一一对应的。两者相比，后者的清晰度明显有了改善。可见，汇编语言一定程度上克服了机器语言难读难改的缺点；同时保持了其编程质量高，占存储空间少，执行速度快的优点。故在程序设计中，对实时性要求较高的地方，如过程控制等，仍经常采用汇编语言。但汇编语言面向机器，使用汇编语言编程需要直接安排存储，规定寄存器和运算器的动作次序，还必须知道计算机对数据约定的表示（定点、浮点、双精度）等。这对大多数人员来说，都不是一件简单的事情。此外，汇编语言还是依赖于机器，不同的计算机在指令长度、寻址方式、寄存器数目、指令表示等都不一样，这样使得汇编程序不仅通用性较差，而且可读性也差，这导致了高级语言的出现。

用汇编语言编写的程序，必须翻译成计算机所能识别的机器语言后，才能被计算机执行。

（3）高级语言

高级语言是由表达各种意义的英文单词和数学公式按照一定的语法规则来编写程序的语言，

如用“+”表示加法、用“-”表示减法等。

高级语言是更接近于自然语言或数学语言的程序设计语言，便于学习和记忆。例如，上述计算 A=15+10 的 BASIC 语言程序如下：

```
A=15+10              '15 与 10 相加的结果放入 A 中
PRINT  A             '输出 A
END                  '程序结束
```

可见，高级语言编写的程序非常直观明了。

高级语言彻底摆脱了依赖于机器硬件的指令系统，是面向应用的计算机语言。它不再依赖于具体的计算机，用高级语言编写的程序可以在不同的计算机上使用，程序具有可移植性。编程时用户不再考虑计算机的内部结构和硬件环境，可以集中精力考虑算法和数据结构，因此编程效率大大提高。

高级语言不能直接在计算机上运行，因为计算机只能识别机器语言程序，高级语言编写的源程序必须经过另外的语言处理程序翻译成机器语言程序后才能被机器接受。因此，高级语言程序的执行速度通常比不上机器语言。

翻译程序分为两种，一种是解释系统，另一种是编译系统。解释系统是对高级语言编写的程序翻译一句执行一句；而编译系统是将高级语言编写的程序文件全部翻译成机器语言，生成可执行文件以后再执行。高级语言几乎在每一种机器上都有自己的翻译程序。C 语言的翻译程序属于编译系统。

高级语言可分为 3 类：面向过程的语言、面向问题的语言和面向对象的语言。

（1）面向过程的语言致力于用计算机能够理解的逻辑来描述需要解决的问题和解决问题的具体方法、步骤。在程序中不仅要告诉计算机“做什么”，还要告诉计算机“如何做”，即在程序中要详细描述用什么动作加工什么数据，即解题的过程和细节。面向过程的语言有 FORTRAN、BASIC、PASCAL、C 等。

（2）面向问题的语言又称非过程化的语言或第四代语言（4GLS）。用面向问题的语言解题时，不必关心问题的求解算法和求解的过程，只需指出问题是要计算机做什么，数据的输入和输出形式，就能得到所需结果。面向问题的语言是采用快速原型法开发应用软件的强大工具，能够快速地构造应用系统，从而大大地提高了软件开发效率。它与数据库的关系非常密切，能够对大型数据库进行高效处理。目前应用最广泛的面向问题的语言有 SQL 等。

（3）面向对象的语言是将客观事物看作具有属性和行为的对象，通过抽象找出同一类对象的共同属性和行为，形成类。通过类的继承与多态可以很方便地实现代码重用，这大大地提高了程序的复用能力和程序开发效率。面向对象语言已是程序语言的主要研究方向之一。面向对象的语言有 C++、Java、Visual Basic 等。

1.1.2　程序设计方法

简单地说，程序设计就是用计算机语言编写程序的过程。学习计算机语言的目的就是利用该语言工具设计出可供计算机运行的程序。通常，程序设计是很讲究方法的，程序设计方法是影响程序设计成败以及程序设计质量的重要因素之一，C 语言主要采用结构化程序设计方法。

结构化程序设计是荷兰学者狄克斯特拉（E.W.Dijkstra）在 1969 年提出的一种程序设计方法，它规定了一套如何进行程序设计的准则，采用了自顶向下逐步求精的分析设计方法、分而治之的分割技术和模块化的组织结构，使得设计的程序具有合理的结构、易读、易调试，容易保证其正确性。

结构化程序设计方法具有以下特点：

（1）自顶向下

（2）逐步细化

（3）模块化设计

（4）结构化编码

结构化程序设计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，这样使完成每一个模块的工作变得单纯而明确，为设计一些较大的软件打下了良好的基础。

由于模块相互独立，因此在设计其中一个模块时，不会受到其他模块的牵连，因而可将原来较为复杂的问题化简为一系列简单模块的设计。模块的独立性还为扩充已有的系统、建立新系统带来了不少的方便，因为我们可以充分利用现有的模块作积木式的扩展。

按照结构化程序设计的观点，任何模块都可以通过3种基本程序结构：顺序结构、选择结构和循环结构组合来实现。3种基本结构应具有如下良好的特性：

（1）只有一个入口，即每个模块与外部联系只有单一的入口；

（2）只有一个出口，即每个模块与外部联系只有单一的出口；

（3）无死语句，即不存在永远都执行不到的语句；

（4）无死循环，即不存在永远都执行不完的循环。

每个模块根据其功能先编写程序框架，逐步深入，直到精确地编写出每一个程序结构，精确地编写每一条语句。

完成编程后，应该检查每条语句、每个程序结构的逻辑及每个模块的功能是否正确，直到检查整个程序是否达到问题的要求，通过编辑、编译、连接运行、调试检验程序是否达到精度要求。

1.1.3 程序设计的基本过程

程序设计是人们根据要解决的实际问题，提出相应的需求，在此基础上设计数据结构和算法，然后再编写相应的源程序代码并测试该代码运行的正确性，通过反复调试直到能够得到正确的运行结果为止，最后整理设计文档的全过程。较小规模的程序设计由一个程序员完成，大型程序设计是由多个程序员分工，共同协作完成的，因此必须经过多种测试，并详细整理设计文档。一般来讲，这个过程应当按图1-1所示的步骤进行。

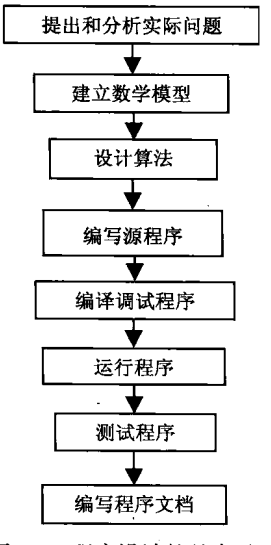


图1-1 程序设计的基本过程

1. 提出和分析实际问题

提出实际需求的用户往往不具备太多的计算机知识，而程序设计人员可能又不具备用户的专业知识，因此，在程序开发初期首先必须由用户和程序设计人员一起对实际问题进行分析，充分理解用户的要求，明确哪些要求可以实现，哪些要求不能实现或需经一定的处理才能实现，确定开发的总目标，提出开发的任务和要求，从中获得必要的输入数据，明确问题要求做什么，需要什么样的数据输出。

分析问题的要求、弄清问题的性质、发现问题的特点、确定解决问题的目标能够使我们采取有效的方法解决问题。

2. 建立数学模型

要用计算机解决实际问题，首先要用理想模型模拟实际问题。理想模型是从实体中抽象出来并能用数学表达式精确定义的实体。建立数学模型的过程就是把错综复杂的实际问题进行简化抽象，用数

学公式来描述实际问题的运动和变化过程。例如，用一些数学方程来描述人造卫星的飞行轨迹等。

在建立数学模型的过程中，分析问题的要求、确定解决问题的目标、明确问题的输入数据和输出信息、理解问题的约束限制条件，是选择制定数学模型、建立数学模型的关键步骤，是解决问题的关键所在。

3. 设计算法

一般来说，从实际问题抽象出数学模型通常是有关领域的专业工作者的任务，计算机工作人员只起辅助作用。程序设计人员的工作，最关键的一步就是设计算法。算法是指为解决某一特定问题而进行一步一步有穷的操作过程，是一组规则的集合，可以用流程图来表示算法。编写程序代码之前，设计好算法，画出流程图，往往会起到事半功倍的效果。

算法可以分为两大类：数值计算算法和非数值计算算法。数值计算算法的目的是求数值解，其特点是少量的输入、输出，复杂的运算，如求高次方程的根、求函数的定积分等。非数值计算算法的目的是对数据的处理，其特点是大量的输入、输出，简单的运算，例如，对数据的排序、查找等算法。

一个好的算法应当具有以下特性。

（1）有穷性：一个算法应包含有限个操作步骤。也就是说，在执行若干个操作步骤之后，算法将结束，并且每一步都在合理的时间内完成。

（2）确定性：算法中每一条指令必须有确切的含义，不能有二义性，并且对于相同的输入必然有相同的执行结果。

（3）可行性：一个算法是可行的，即算法中指定的操作，都可以通过已实现的基本运算执行有限次来实现。

（4）输入：一个算法一般有零个或多个输入。在计算机上实现的算法，通常是用来处理数据对象的，在大多数情况下这些数据是需要通过输入来得到的。

（5）输出：一个算法一般有一个或多个输出，算法的目的是为了求“解”，这些“解”只有通过输出才能得到。

描述算法的常用工具是流程图，也称程序框图，流程图是算法的图形描述，它用一组图形符号来表示各种操作，它往往比程序更直观，能较清晰地表达各种操作之间的逻辑关系，容易阅读和理解。常用的流程图符号如图 1-2 所示。

图 1-2　常用流程图符号

4. 编写源程序

如果算法正确，将它转换为任何一种高级语言程序，并不困难，这一步通常称为编码。

现在的程序设计语言一般都是一个集成开发环境，自带编辑器，方便编辑程序。编写好的程序代码通过编辑器输入到计算机内，利用编辑器可对输入的程序代码进行复制、删除、移动等编辑操作，然后以文件（源程序）形式保存。

5. 编译调试程序

源程序必须通过编译程序将源程序翻译成目标程序，这期间编译器对源程序进行语法结构检查，找出在程序编制过程中存在的语法错误，加以修改。这是一个重复进行的过程，需要反复调试。

6. 运行程序

将编译源程序生成的目标程序和程序中所需的系统中固有的目标程序模块（如调用的标准函

数、执行的输入/输出操作的模块）连接后生成可执行文件。运行该程序，检查程序输出结果是否正确，如发现错误，检查程序的逻辑是否正确，主要解决算法设计错误。

7. 测试程序

测试程序是对照问题的要求，通过让程序试运行一组数据，分析输出的结果，检查是否达到问题要求的功能和精度要求，输出信息的格式是否符合要求。这组测试数据应是以任何程序都是有错误的前提精心设计出来的，称为测试用例。

测试有黑盒测试和白盒测试两种方法。对于不同的测试方法有不同的测试用例。

（1）黑盒测试：也称为功能测试或数据驱动测试。它把程序看成一个黑盒子，完全不考虑程序的内部结构和处理过程，只对程序的接口进行测试，即检查程序是否能适当地接受输入数据并产生正确的输出信息。实际上目前有些软件开发商推出的软件 β 版（即测试版），免费提供给用户使用，从使用角度找出软件的问题，即属于黑盒测试方法。

（2）白盒测试：把程序看成一个透明的白盒子，也就是完全了解程序的内部结构和处理过程。这种方法按照程序内部的逻辑来测试，检验程序中的每条通路是否正确工作。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试一般由计算机专业人员进行。

例如对求解一元两次方程根：$ax^2+bx+c=0$，测试用例可为：

$a=0$，$b=0$，$c=0$，$b^2-4ac\geq 0$，$b^2-4ac<0$

等各种特殊情况时对应输入 a、b、c 的值，观察程序运行的结果，属于白盒测试法；若输入任何 a、b、c 的值包括非法的数据，观察程序运行的结果，属黑盒测试法。

8. 编写程序文档

在程序准确无误后，要认真编写程序文档。文档包括设计要求、设计思路、设计过程、使用的算法、数据结构、输出信息及格式等，在源程序中要用注释语句加上必要的说明。如果没有程序文档，编制的程序过一段时间后自己也看不懂，更不要说给别人看，对此要引以为戒，要学会从学习程序设计开始就养成良好的习惯。

1.2 C 语言及其特点

C 语言是一种得到广泛重视并普遍应用的计算机程序设计语言，也是国际公认的最重要的几种通用程序设计语言之一，它既可用来编写系统软件也可用来编写应用软件。

1.2.1 C 语言的发展过程

C 语言的发展是一个充实和完善的过程。

1972 年，C 语言是由贝尔实验室的 Dennis Ritchie 和 Brian Kernighan 根据 B 语言设计的，而 B 语言又是由一种早期的编程语言 BCPL（Basic Combined Programming Language）发展演变而来的。BCPL 的根源可以追溯到 1960 年的 ALGOL 60（Algol Programming Language），ALGOL 60 是一种面向问题的高级语言，离硬件较远。1963 年，英国剑桥大学推出 CPL（Combined Programming Language），CPL 修改了 ALGOL 60，使其能够直接作较低层次的操作。1967 年英国剑桥大学的 Martin Richards 对 CPL 做了改进，推出了 BCPL 语言。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，又做了进一步简化，设计出了很简单的而且很接近硬件的 B 语言（取 BCPL 的第一个字母），并用 B 语言写了第一个 UNIX 操作系统，在 PDP-7 上实现。1971 年在 PDP-11/20

上实现了 B 语言，并写了 UNIX 操作系统。但 B 语言过于简单，功能有限。1972 年至 1973 年间，Dennis Ritchie 在 B 语言的基础上增加了类型（Datatype）和结构（Structure），设计出了 C 语言（取 BCPL 的第二个字母）。C 语言既保持了 BCPL 和 B 语言的优点（精练，接近硬件），又克服了它们的缺点（过于简单，数据无类型等）。

最初的 C 语言是为描述和实现 UNIX 操作系统提供的一种工具语言。但 C 语言并没有被束缚在任何特定的硬件或操作系统上，它具有良好的可移植性。1977 年出现了不依赖于具体机器的 C 语言编译文本——《可移植 C 语言编译程序》，用该程序编写的 UNIX 系统迅速在各种机器上实现，UNIX 系统支持的 C 语言也被移植到相应的计算机上。C 语言和 UNIX 系统在发展过程中相辅相成，得到了广泛应用，使它先后被移植到各种大、中、小、微型计算机上。

以 1978 年发表的第七版本 UNIX 系统中的 C 语言编译程序为基础，B.W.Kernighan 和 D.M.Ritchie 合著了《The C Programming Language》。这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础，被称为标准 C 语言。1983 年，美国国家标准化协会（ANSI）根据 C 语言问世以来的各种版本对 C 语言的发展和扩充制定了新的标准，称为 ANSI C。1990 年，C 语言成为国际标准化组织（ISO）通过的标准语言。本书以 ANSI C 为基础，以 Borland 公司的 Turbo C 2.0 为扩充。书中的程序全部在 Turbo C 2.0 集成环境中调试通过。

1.2.2 C 语言的特点

一种语言之所以能存在和发展，并具有生命力，总是有不同于其他语言的特点。C 语言也是如此，它的特点是多方面的，人们从不同的角度可总结出众多的特点，但全面考虑可归纳为以下几点。

（1）C 语言是比较低级的语言。有人把 C 语言称为高级语言中的低级语言，也有人称它是中级语言。它具有许多通常只有像汇编语言才具备的功能，如位操作、直接访问物理地址等，这使 C 语言在进行系统程序设计时显得非常有效，而过去系统软件通常只能用汇编语言编写。事实上，C 语言的许多应用场合是汇编语言的传统领地，现在用 C 语言代替汇编语言，使程序员得以减轻负担，提高效率，而且写出的程序具有更好的可移植性。

C 语言具有更多的接近硬件操作的功能，而不提供直接处理复合对象的功能。如作为整体看待的字符串、数组等操作，这些较高级的功能必须通过调用函数来完成。这看起来是个缺陷，但这样使得语言规模小，更容易说明，学习起来也快。比如说，C 语言只有 32 个关键字，而一些微机上的 BASIC 语言，关键字多达 100 个以上。

（2）C 语言是结构化的程序设计语言。C 语言主要结构成分是函数，函数允许一个程序中的各任务分别定义和编码，使程序模块化。C 语言还提供了多种结构化的控制语句，如用于循环的 for、while、do-while 语句，用于判定的 if-else、switch 语句等，十分便于采用自顶向下、逐步细化的结构化程序设计技术。因此，用 C 语言编制的程序容易理解、便于维护。

（3）C 语言具有丰富的运算能力。在 C 语言中除了一般高级语言使用的算术运算及逻辑运算功能外，还具有独特的以二进制位（bit）为单位的位与、位或、位非以及移位操作等运算。并且 C 语言具有如 a++、b—等单项运算和+=、-=等复合运算功能。

（4）C 语言数据类型丰富，具有现代化语言的各种数据类型。C 语言的基本数据类型有整型（int）、浮点型（float）、字符型（char）。在此基础上按层次可产生各种构造类型，如数组、指针、结构体、共用体等。同时还提供了用户自定义数据类型。用这些数据类型可以实现复杂的数据结构，如栈、链表、树等。因此，C 语言具有较强的数据处理能力。

（5）C 语言具有预处理能力。在 C 语言中提供了#include 和#define 两个预处理命令实现对外

部文件的包含以及对字符串的宏定义。同时还具有#if ~ #else 等条件编译预处理语句。这些功能的使用提高了软件开发的工作效率并为程序的组织和编译提供了便利。

（6）C 语言可移植性好。目前，C 语言在许多机器上实现，大部分是由 C 语言编译移植得到的。C 编译程序的可移植性，也就使 C 语言程序便于移植。

C 语言的优点很多，但也有一些不足。如语法限制不太严格、类型检验太弱、不同类型数据转换比较随便，这就要求程序员对程序设计的方法和技巧更熟练，以保证程序的正确性。

总之，C 语言已成为国内外广泛使用的一种编程语言，并且非常适合于程序设计语言课程的教学工作中。

1.3 C 语言程序的结构特点与书写规则

学习 C 语言的关键是要把握程序设计的思想，反复实践，培养自己独立编写程序的能力。下面通过几个简单的 C 程序实例介绍 C 语言程序的基本结构和编写方法，使读者对 C 语言程序有一个初步的认识。

1.3.1 C 语言程序的基本结构

C 语言是函数型语言，函数是构成 C 语言程序的基本单位。下面我们通过几个实例来分析 C 语言程序的组成和结构。

例 1.1 编写 C 语言程序，实现在屏幕上显示字符串“This is a C program.”。

具体程序如下：

```
#include "stdio.h"                          /* 包含有关标准库的信息 */
void main(void)
{
  printf("This is a C program.\n");
}
```

运行结果：

```
This is a C program.
```

说明：

（1）这是一个完整的 C 语言程序。其中以“#”开始的语句是预处理命令。这些命令是在编译系统翻译代码之前需要由预处理程序处理的语句。本例中的#include "stdio.h"语句是请求预处理程序将文件 stdio.h 包含到程序中来，作为程序的一部分。文件 stdio.h 为输入和输出提供支持，在本程序中的 printf（"This is a C program.\n"）；语句的执行需要 stdio.h 的支持，没有它，程序将不能通过编译系统的翻译。

（2）/* 包含有关标准库的信息 */部分为注释语句。注释只是给人看的，不影响程序的执行，只是为了帮助读者阅读和理解程序。

（3）void main（void）表明以下是一个 C 语言程序的主函数，每一个 C 语言程序都必须有一个 main 函数。用{}括起来的部分是函数体。

（4）printf()是 C 语言提供的标准输出库函数，它的功能是将一对双引号中的内容输出到标准输出设备显示器上。“\n”是换行符，即在输出“This is a C program.”后回车换行。printf()后的分号是语句的结束符，C 语言程序的每一个语句都以分号“;”终止。

例 1.2　编写程序，计算两个整数之和，由主函数独立完成。

具体程序如下：

```
#include "stdio.h"
void main(void)
{
  int a,b,sum;                          /* 定义 a，b 和 sum 3 个整型变量 */
  a=10;                                 /* 为 a 赋值 */
  b=20;                                 /* 为 b 赋值 */
  sum=a+b;                              /* 将变量 a 与 b 相加的和值赋给变量 sum */
  printf("sum=a+b=%d\n",sum);           /* 将 sum 变量的值输出到屏幕上*/
}
```

运行结果：

```
sum=a+b=30
```

说明：

（1）int a,b,sum；是数据类型说明语句，把 a、b 和 sum 定义为整型（int）变量，int 是类型说明符。值得注意的是，所有 C 语言程序中的变量，在使用之前都要定义其数据类型，以便在程序运行时分配相应的存储空间。

（2）sum=a+b；是一个计算表达式，表示把等号右边的运算结果赋给左侧的变量 sum。

（3）printf（"sum=a+b=%d\n",sum）；是输出语句，它先在新的一行上输出字符串“sum=a+b=”，然后按十进制整型数据格式（%d）输出变量 sum 的值，并使光标移到下一行。

其中“%d”是输入/输出的格式字符串，用来指定输入/输出时的数据类型和格式。执行输出时，此位置会被后面对应的表达式的值取代，即输出 sum 的值。

例 1.3　求两个整数之和，由主函数和一个用户自定义函数合作完成。

具体程序如下：

```
#include "stdio.h"
void main(void)                         /* 主函数 */
{
   int a,b,sum;                         /* 定义 a，b 和 sum 3 个变量 */
   a=10; b=20;                          /* 为 a，b 赋值 */
   sum=add(a,b);                        /* 调用函数 add，将得到的值赋给变量 sum */
   printf("sum=a+b=%d\n",sum);          /* 屏幕输出 sum 变量的值 */
}
int add(int x,int y)                    /* 定义 add 函数和形式参数 x，y */
{
  int z;                                /* 定义 z 变量 */
  z=x+y;                                /* 变量 x 与 y 相加的和赋给 z */
  return(z);                            /* 返回 z 的值，通过 add 带回调用处 */
}
```

运行结果：

```
sum=a+b=30
```

说明：

该程序的功能与例 1.2 相同，只是采用了函数调用来计算两个数的和。此程序由两个函数组成：主函数 main()和被调用的函数 add()。main()函数负责数据的输入和输出；add()函数负责将 x 与 y 相加的和赋给变量 z。

sum=add（a,b）；为函数调用语句。主函数在调用 add()函数时，将实际参数 a 和 b 的值分别传递给 add()函数中的形式参数 x 和 y。add()函数收到数据后，对它们进行计算，将结果放入变量 z 中，并将结果变量 z 的值返回给主函数，由主函数负责输出结果值，即两个函数共同合作完成任务。

例 1.4 从键盘输入两个整数，求其中较小数，并输出结果。

具体程序如下：

```
#include "stdio.h"
int min(int x,int y)                    /* 定义 min 函数，x,y 为形参 */
{                                       /* 函数体开始 */
  int z;                                /* 定义函数中使用的变量 z */
  if(x<y) z=x;                          /* 条件语句：如果 x 小于 y 那么把 x 的值赋给 z */
  else   z=y;                           /* 否则把 y 的值赋给 z */
  return(z);                            /* 将 z 的值返回，通过 min 带回调用处 */
}                                       /* 函数体结束 */
main()                                  /* 主函数 */
{                                       /* 函数体开始 */
  int a,b,c;                            /* 定义函数中使用的 3 个变量 a，b 和 c */
  printf("Input two integers: ");       /* 输出一行提示信息 */
  scanf("%d,%d",&a,&b);                 /* 用标准输入函数输入两个整数，赋值给 a 和 b */
  c=min(a,b);                           /* 调用 min 函数，将结果赋给 c 变量 */
  printf("min=%d",c);                   /* 输出 c 变量的值 */
}                                       /* 函数体结束 */
```

运行输入：

```
Input two integers: 3,5↙
```

运行结果：

```
min=3
```

说明：

当运行上面这个程序时，首先，屏幕上显示一条提示信息：

Input two integers:

要求用户从键盘输入两个整数。如果此时用户输入 3 和 5，即

Input two integers: 3,5↙

这里，符号“↙”表示按一下回车键，以示输入结束。此时屏幕显示运行结果。

本程序包括主函数 main()和被调用的函数 min()。scanf()是 C 语言提供的标准输入库函数，它的功能是把从键盘上输入的数据传送给对应的变量。程序执行 scanf()时，操作员由键盘输入两个整数值分别送给变量 a 和 b。程序执行 c=min(a,b)时，调用 min()函数，将 a 的值送给 x，b 的值送给 y，程序转到 min()函数执行，min()函数中的 if 语句的作用是将 x 变量和 y 变量中的较小值赋给 z 变量。return 语句的作用是将 z 变量的值返回给 min()函数同时程序返回主函数执行，min()函数值再送给 c 变量。最后 printf()将 c 变量的值输出到屏幕。

通过以上几个实例，可以看到 C 语言程序的结构有如下特点：

1. C 语言程序是由函数组成的

C 语言源程序由若干个函数组成，函数是 C 程序的基本单位。组成程序的若干函数中必须有且仅有一个名为 main 的函数。例 1.3 中包含两个函数： main()和 add()。因为在 main()函数中调用 add()函数，所以 main()为主函数，add()为被调用的函数。被调用函数可以是系统提供的库函数