

21世纪高校计算机系列规划教材

计算机程序设计基础 (C语言版)

>>> 朱二连 等编著

Computer Programming



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

“十二五”普通高等教育本科国家级规划教材

计算机组成设计基础

（C语言版）

王冬冬 王冬 王冬 王冬 王冬 王冬 王冬 王冬 王冬 王冬

清华大学出版社

（北京）

（上海）

（广州）

（深圳）

（香港）

（澳门）

（台北）

（新加坡）

（马尼拉）

（曼谷）

（雅加达）

（泗水）

（巴达维亚）

（巨港）

（棉兰）

（三寶壟）

（万隆）

（日惹）

（梭罗）

（三寶壟）

（三寶壟）

21 世纪高校计算机系列规划教材

计算机程序设计基础

(C 语言版)

朱二连 等编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

计算机程序设计是高等院校计算机重要基础课程之一。学习程序设计重点究竟是学习工具还是学习方法设计正是本书解决的问题。因此,本书在结构上突出了以程序设计为中心,以解决实际问题为目的,以语言知识为工具的思想,在解决实际问题过程中对 C 语言的语法规则进行了整理和提炼,深入浅出地介绍了它们在程序设计中的应用;在内容上注重知识的完整性,以适合初学者的需求;在写法上追求循序渐进,通俗易懂。

全书分为 7 章,主要内容包括:程序设计基础知识以及 C 语言概述,算法描述、基本程序结构与 C 语言控制语句,较少数据的程序设计以及 C 语言的基本输入/输出和程序设计风格,结构化程序设计方法、较多数据的程序设计以及数组和结构体的应用、大量数据的程序设计以及文件的应用,此外还介绍了 C 语言特色如 C 语言运算符与表达式计算、指针及其应用等。书后还有 7 个附录,以供读者学习时查阅。

本教材配有教学参考书《计算机程序设计基础(C 语言版)实验教程》,以方便读者上机操作。本书适合作为高等院校理工类非计算机专业学生的计算机程序设计教材,也可以作为计算机专业学生的计算机程序设计的入门教材,还可以作为科技人员和相关人员的自学参考书。

图书在版编目(CIP)数据

计算机程序设计基础: C 语言版 / 朱二连等编著. —北京: 中国铁道出版社, 2009. 3
(21 世纪高校计算机系列规划教材)
ISBN 978-7-113-09707-3

I. 计… II. 朱… III. ①程序设计—高等学校—教材②C 语言—程序设计—高等学校—教材 IV. TP311.1 TP312

中国版本图书馆 CIP 数据核字(2009)第 026854 号

书 名: 计算机程序设计基础(C 语言版)
作 者: 朱二连 等编著

策划编辑: 秦绪好 杨 勇

责任编辑: 王占清

编辑部电话: (010) 63583215

编辑助理: 郝霁江 王 彬

封面设计: 付 巍

封面制作: 白 雪

责任印制: 李 佳

出版发行: 中国铁道出版社(北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 北京新魏印刷厂

版 次: 2009 年 3 月第 1 版 2009 年 3 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 16 字数: 377 千

印 数: 5 000 册

书 号: ISBN 978-7-113-09707-3/TP·3156

定 价: 24.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

随着计算机技术的发展与普及,计算机已经成为各行各业最基本的工具之一,而且正迅速地进入人类生活的各个领域。C语言作为国际上广泛流行的通用程序设计语言,在计算机的研究和应用中已展现出强大的生命力。C语言兼顾了诸多高级语言的特点,是一种典型的结构化程序设计语言,它处理能力强,使用灵活方便,应用面广,具有良好的可移植性,既适合于计算机专业人员编写系统软件,又适合于应用开发人员编写应用软件。所以长久以来广泛流行,经久不衰。

但是,由于C语言比其他计算机语言的学习难度要大,尤其是复杂表达式、变量的作用域和生命期、指针的应用以及位运算等内容,对初学计算机语言的人来说,有很大难度。所以,在绝大部分高校中,以前总是先开设其他计算机语言课,然后再学习C语言。自从20世纪90年代后期,随着计算机新技术的不断涌现,高校计算机基础教学内容一再增加,非计算机专业本科生的计算机语言只开设一门课程,“C语言程序设计”就被选为非计算机理工科专业第一门(也是唯一一门)本科生的程序设计必修课。

几年来,每次给学生讲授C语言课程时,都会遇到同样的问题,就是学生在看书或者上机做实验时,都把主要精力和时间花在死抠复杂表达式和格式输入/输出等方面,没有把主要精力用在编写计算程序上,而且现在的教材也都是这样安排的。我们就一直在想,作为程序设计的入门教材,应该介绍程序设计的基本概念和基本方法,在此前提下,再由浅入深地介绍C语言的相关内容。因此有了这样一个构思:编写一本能够让大多数人可以在很短时间内掌握程序设计思想、程序的基本结构并且用C语言基本语句来描述程序,而且还能够自己学会用其他计算机语言来编写计算机程序的教材,供学生和广大编程爱好者自学选用。

本书作为程序设计教材的创新,实现了从以计算机语言为主线的体系结构向以问题为主线的体系结构上的转变,把程序设计的学习从语法知识学习提高到解决问题的能力培养上、从花大量时间死抠复杂表达式和格式输入/输出等方面解脱出来而把主要精力用在编写计算程序上。本教材在结构上突出了以程序设计为中心,以语言知识为工具的思想。在内容上注重知识的完整性,以适合初学者的需求。对C语言的语法规则进行了整理和提炼,深入浅出地介绍了它们在程序设计中的应用和用计算机解决问题的方法,把其他教科书中对C语言中过于“技巧性”问题的讨论与介绍安排在后面章节,供有兴趣和有需要的人可以进一步继续深入学习。

全书共分7章,组织方式为以解决问题为主线,在解决问题时需要用到C语言的哪些内容就来介绍相关内容,力图把程序设计的学习从语法知识学习提高到解决问题的能力培养上。如第1章介绍了程序的相关概念、C语言基本语法元素以及C语言程序的结构特

点与开发环境等。第2章介绍C语言的基本数据类型、基本程序结构和基本控制语句(顺序、选择、循环等控制语句)以及实例。第3章围绕编写处理较少数据的程序,介绍了C语言的基本输入/输出,以及模块化程序设计与C语言函数及其调用等内容。此外还介绍了程序设计相关的概念,如算法及其描述、程序设计风格等内容。第4章围绕编写处理较多数据的程序,介绍了数组在处理较多数据的程序设计中的应用,特别是用结构体数组来处理较多较复杂数据的程序设计方法。第5章围绕编写处理大量数据的程序,介绍了文件在处理大量数据的程序设计中的应用。第6章主要介绍了C语言特色中有关数据类型转换与混合运算、数据类型与数据存储结构以及变量的作用域与存储方式。此外还详细介绍了C语言丰富的运算符及其结合性以及运算符与表达式,包括算术、赋值、关系、逻辑、求字节、逗号、条件、位运算、指针运算等。第7章主要介绍C语言最具特色的指针概念在程序设计中的应用,包括指针与数组、指针与函数、指针与字符串、指针与结构体、指针与链表等内容。此外,本章还介绍了C语言中的编译预处理,包括宏定义、文件包含、条件编译等内容。

全书各章配有习题,并有配套的上机指导书。本书适合作为高等院校非计算机理工类专业学生的计算机语言教材,也可以作为高等院校计算机专业本、专科低年级学生学习计算机语言的入门教材。本书中介绍的C语言,覆盖了国家教育委员会考试中心的《全国计算机等级考试考试大纲》中的二级考试大纲“C语言程序设计考试要求”。因此也适合参加二级C语言考试的人员选用。

本书由石家庄经济学院朱二连,海南大学信息学院黄筱燕、张峰,长春一汽启明信息技术股份有限公司NES项目组刘涛,石家庄经济学院庞南雁等编著。在本书的策划和出版过程中,一直得到学校各级领导的大力支持,许多从事计算机基础教学工作的同仁也给予了关心和帮助。在本书的编写过程中,得到了石家庄经济学院信息工程学院计算机基础教研室计算机程序设计课程组陈巖瑛、郭秀敏、赵占芳、王楠、马立霄、赵洋以及关文革等老师的支持和帮助。此外,本书的出版还得到了中国铁道出版社领导和相关编辑的大力支持和帮助,在此一并表示最真诚的谢意。

经过几年的积累,终于完成了书稿。但由于编者水平有限,书中难免有疏漏之处,恳请专家和读者批评指正。

朱二连

2009年1月6日于石家庄

第 1 章 C 语言概述.....	1
1.1 程序与程序设计语言	1
1.1.1 程序设计基本概念	1
1.1.2 程序设计语言	1
1.2 C 语言概述	3
1.2.1 C 语言发展	3
1.2.2 简单 C 程序介绍	4
1.2.3 C 程序结构特点	6
1.3 C 语言基本语法元素	6
1.3.1 C 语言符号集	6
1.3.2 C 语言中的词汇	6
1.4 C 语言程序开发环境	8
1.4.1 Turbo C 2.0 介绍	8
1.4.2 源程序输入	9
1.4.3 C 程序的编译、连接与运行	9
本章小结	10
习题	10
第 2 章 C 语言基础与程序基本结构	11
2.1 C 语言的数据与数据类型	11
2.1.1 C 语言中的数据	11
2.1.2 C 语言数据类型	12
2.2 C 语言运算符与表达式	15
2.2.1 C 语言的运算符	15
2.2.2 C 语言的表达式	17
2.3 C 语言的语句	19
2.3.1 简单语句	20
2.3.2 复合语句	20
2.3.3 空语句	20
2.3.4 程序结构控制语句	20
2.4 程序基本结构与 C 程序实例	21
2.4.1 程序的基本结构	21
2.4.2 顺序结构程序示例	22
2.4.3 分支结构控制语句及示例	24
2.4.4 循环结构控制语句及示例	30

本章小结	34
习题	35
第 3 章 较少数据的程序设计	40
3.1 算法与程序设计	40
3.1.1 算法的概念	40
3.1.2 算法的描述方法	42
3.1.3 结构化程序设计方法	44
3.1.4 程序设计风格	45
3.2 数据输入与输出	46
3.2.1 格式化输出函数 printf()	47
3.2.2 格式化输入函数 scanf()	49
3.2.3 字符输入/输出函数 getchar()/putchar()	50
3.3 较少数据的程序设计实例	51
3.4 函数与模块化编程	67
3.4.1 标准库函数	67
3.4.2 用户定义函数	68
3.4.3 模块化程序设计实例	72
3.5 程序的书写风格	76
本章小结	77
习题	78
第 4 章 较多数据的程序设计	81
4.1 数组及其元素控制	81
4.1.1 一维数组	82
4.1.2 二维数组	84
4.1.3 数组作为函数参数的应用	87
4.1.4 应用程序举例	88
4.2 字符数组与字符串	91
4.2.1 字符数组的定义与初始化	92
4.2.2 字符串的输入与输出	92
4.2.3 字符串的应用	93
4.2.4 字符串处理函数	95
4.3 结构体	95
4.3.1 结构体类型变量的定义和使用	95
4.3.2 结构体数组的定义和初始化	99
4.3.3 结构体数组的使用	100
4.3.4 结构体作为函数参数的使用	101
4.4 较多数据的程序设计实例	102

本章小结	107
习题	108
第 5 章 大量数据的程序设计	110
5.1 文件概述	110
5.1.1 文件	110
5.1.2 数据文件的存储形式	111
5.1.3 文件的存取方法	112
5.1.4 文件的类型标识与文件类型变量	112
5.2 文件的基本操作	112
5.2.1 文件的打开与关闭操作	113
5.2.2 文件结束检测操作	114
5.3 文件的读/写操作	115
5.3.1 字符方式的读/写操作	115
5.3.2 字符串方式的读/写操作	116
5.3.3 格式化的读/写操作	118
5.3.4 块方式的读/写操作	119
5.3.5 应用举例	121
5.4 文件的定位操作	123
5.4.1 rewind()函数	123
5.4.2 fseek()函数	123
5.4.3 ftell()函数	124
5.4.4 应用举例	124
5.5 大量数据的程序设计实例	126
本章小结	127
习题	127
第 6 章 C 语言特色介绍 (1)	129
6.1 数据类型转换与混合运算	129
6.1.1 再论 C 语言数据类型	129
6.1.2 数据类型转换与混合运算	130
6.1.3 数据类型定义符 typedef	131
6.2 数据类型与数据存储结构	132
6.2.1 基本数据类型及其存储	132
6.2.2 构造类型及其存储	140
6.2.3 指针类型及其存储	147
6.3 变量的作用域与存储类别	150
6.3.1 变量的作用域	150
6.3.2 变量的存储类别	153

6.4 再论运算符与表达式	155
6.4.1 运算符及其结合性	155
6.4.2 运算符与表达式	155
本章小结	169
习题	170
第 7 章 C 语言特色介绍 (2)	177
7.1 编译预处理	177
7.1.1 宏定义	177
7.1.2 文件包含	183
7.1.3 条件编译	184
7.2 指针的应用	186
7.2.1 再论指针数据类型	186
7.2.2 指针与函数	188
7.2.3 指针与数组	191
7.2.4 指针与字符串	198
7.2.5 指针与结构体	202
7.2.6 指针与链表	206
本章小结	220
习题	221
附录 A 常用字符与 ASCII 码对照表	227
附录 B C 语言中的关键字	229
附录 C C 语言 9 种控制语句	230
附录 D 运算符的优先级和结合性	231
附录 E Turbo C 常用库函数与头文件	232
附录 F C 程序常见错误	236
附录 G Turbo C 调试程序常见出错提示信息	242
参考文献	246

第 1 章 C 语言概述

自 1946 年世界上第一台电子计算机问世以来, 计算机科学及其应用的发展十分迅猛, 计算机被广泛地应用于人类生产、生活的各个领域, 推动了社会的进步与发展。特别是随着因特网 (Internet) 日益深入千家万户, 传统的信息收集、传输及交换方式正被革命性地改变, 我们已经难以摆脱对计算机的依赖, 计算机已将人类带入了一个新的时代——信息时代。新的时代对于我们的基本要求是: 自觉地、主动地学习和掌握计算机的基本知识和基本技能, 并把它作为自己应该具备的基本素质。

对于使用计算机的人来说, 掌握一门高级语言及其基本的编程技能是必需的。本章主要从程序设计的角度, 介绍有关程序设计的基本概念, 结合 C 语言的发展, 介绍 C 语言程序的基本结构以及 C 语言程序的开发环境等。

1.1 程序与程序设计语言

1.1.1 程序设计基本概念

所谓程序, 实际上就是用计算机语言描述的解决某一个问题的步骤, 是遵循一定语法规则的指令序列 (代码)。人们借助计算机语言, 告诉计算机要处理什么 (即要处理哪些数据) 以及如何处理 (即按什么步骤来处理), 这就是程序设计。将编写好的程序在计算机上运行, 便可使计算机按人们的要求解决特定的问题。

我们知道, 解决一个问题的方法或者步骤不是唯一的, 因此不同用户开发 (编写) 的程序也可能是不一样的。不同的程序有不同的效率, 这涉及程序的优化以及程序所采用的数据结构、算法等方面的问题, 还要将该程序在计算机上测试直到运行结果正确。所以, 程序设计应该遵循一定的方法和原则, 还要养成良好的程序设计风格, 才能使设计的程序具有可靠性、可读性以及可维护性等, 这些内容在以后会详细介绍。

1.1.2 程序设计语言

要完成程序设计, 自然离不开程序设计语言。像人类语言一样, 计算机程序设计语言也具有一定的规则: 固定的语法格式、特定的意义 (语意) 和使用环境 (语境), 这些规则比人类语言更加严格, 因为计算机是不能分析语言的多义性的。

一台计算机是由硬件系统和软件系统两大部分构成的, 硬件是物质基础, 而软件可以说是计算机的灵魂, 没有软件, 计算机就是一台“裸机”, 是什么也不能干的; 有了软件, 才能灵动起来

并成为一台真正的“电脑”。所有的软件都是用计算机语言编写的。计算机程序设计语言的发展,经历了从机器语言、汇编语言到高级语言的历程。

1. 机器语言

电子计算机所使用的是由“0”和“1”组成的二进制数,二进制是计算机语言的基础。计算机发明之初,人们只能降贵纡尊,用计算机的语言去命令计算机工作,一句话,就是写出一串串由“0”和“1”组成的指令序列交由计算机执行,这种语言就是机器语言。使用机器语言是十分痛苦的,特别是在程序有错需要修改时,更是如此。而且,由于每台计算机的指令系统往往各不相同,所以在一台计算机上执行的程序,要想在另一台计算机上执行,必须另编程序,造成了重复工作。但由于使用的是针对特定型号计算机的语言,故而运算效率是所有语言中最高的。机器语言是第一代计算机语言。

2. 汇编语言

为了减轻使用机器语言编程的痛苦,人们进行了一种有益的改进:用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串,例如用 ADD 代表加法,MOV 代表数据传递等,这样一来,人们很容易读懂并理解程序在干什么,纠错及维护都变得方便了,这种程序设计语言就称为汇编语言,即第二代计算机语言。然而计算机是不认识这些符号的,这就需要有一个专门的程序,专门负责将这些符号翻译成二进制数的机器语言,这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于计算机硬件,移植性不好,但效率仍十分高,针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精炼且质量高,所以至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的痛苦经历中,人们意识到,应该设计一种这样的语言,这种语言接近于数学语言或人的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有机器上通用。经过努力,1954年,第一个完全脱离计算机硬件的高级语言——FORTRAN 问世了,50多年来,共有几百种高级语言出现,具有重要意义的有几十种,影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、SNOBOL、PL/I、Pascal、C、PROLOG、Ada、C++、VC、VB、Delphi、Java 等。

高级语言的发展也经历了从早期语言到结构化程序设计语言,从面向过程到非过程化程序语言的过程。相应地,软件的开发也由最初的个体手工作坊式的封闭式生产,发展为产业化、流水线式的工业化生产。20世纪60年代中后期,软件越来越多,规模越来越大,而软件的生产基本上是各自为战,缺乏科学规范的系统规划与测试、评估标准,其恶果是大批耗费巨资建立起来的软件系统,由于含有错误而无法使用,甚至带来巨大损失,软件给人的感觉是越来越不可靠,以致几乎没有不出错的软件。这一切极大地震动了计算机界,史称“软件危机”。人们认识到:大型程序的编制不同于写小程序,它应该是一项新的技术,应该像处理工程一样处理软件研制的全过程。程序的设计应易于保证正确性,也便于验证正确性。1969年,提出了结构化程序设计方法,1970年,第一个结构化程序设计语言——Pascal 语言出现,标志着结构化程序设计时期的开始。

20世纪80年代初,在软件设计思想上,又发生了一次革命,其成果就是面向对象的程序设计。在此之前的高级语言,几乎都是面向过程的,程序的执行是流水线似的,在一个模块被执行完成前,人们不能干别的事,也无法动态地改变程序的执行方向。这和人们日常处理事物的方式

是不一致的, 对人们而言是希望发生一件事就处理一件事, 也就是说, 不能面向过程, 而应是面向具体的应用功能, 也就是对象 (object)。其方法就是软件的集成化, 如同硬件的集成电路一样, 生产一些通用的、封装紧密的功能模块, 称之为软件集成块, 它与具体应用无关, 但能相互组合, 完成具体的应用功能, 同时又能重复使用。对使用者来说, 只关心它的接口 (输入量、输出量) 及能实现的功能, 至于如何实现的, 那是它内部的事, 使用者完全不用关心, C++、VB、Delphi 就是典型代表。

高级语言的下一个发展目标是面向应用, 即只需要告诉计算机你要干什么, 计算机就能自动生成算法, 自动进行处理, 这就是非过程化的程序语言。

1.2 C语言概述

几十年来, 计算机程序设计语言发展非常迅速, 新的程序设计语言层出不穷, 其功能也越来越强大。在种类繁多的计算机程序设计语言中, C语言是最具代表性的计算机程序设计语言之一, 因为不仅可以用它开发应用软件, 还可以用于开发系统软件。

1.2.1 C语言发展

C语言是在20世纪70年代初问世的。1978年由美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。同时由B.W.Kernighan和D.M.Ritchie合著了著名的 *THE C PROGRAMMING LANGUAGE* 一书, 通常简称为 *K&R*, 也有人称之为 *K&R* 标准。但是, 在 *K&R* 中并没有定义一个完整的标准C语言, 后来由美国国家标准学会在此基础上制定了一个C语言标准, 于1983年发表, 通常称之为 ANSI C, 它是当时最优秀的程序设计语言这一。

早期的C语言主要是用于UNIX系统。由于C语言的强大功能和各方面的优点逐渐为人们认识, 到了20世纪80年代, C开始进入其他操作系统, 并很快在各类大、中、小和微型计算机上得到了广泛的使用。

C语言是一种结构化语言。它层次清晰, 便于按模块化方式组织程序, 易于调试和维护。C语言的表现能力和处理能力极强。它不仅具有丰富的运算符和数据类型, 便于实现各类复杂的数据结构, 还可以直接访问内存的物理地址, 进行位 (bit) 一级的操作。由于C语言实现了对硬件的编程操作, 因此C语言集高级语言和低级语言的功能于一体。它既可用于系统软件的开发, 也适合于应用软件的开发。此外, C语言还具有效率高、可移植性强等特点。因此, 它被广泛地移植到了各种类型计算机上, 从而形成了多种版本的C语言。

目前最流行的C语言有以下几种:

- Microsoft C 或称 MS C。
- Borland Turbo C 或称 Turbo C。
- AT&T C。

这些C语言版本不仅实现了ANSI C标准, 而且在此基础上各自做了一些扩充, 使之更加方便、完美。

在C语言的基础上, 1983年又由贝尔实验室的Bjarne Stroustrup推出了C++。C++进一步扩充和完善了C语言, 成为一种面向对象的程序设计语言。C++最为常用的版本是 Borland C++ 4.5,

Symantec C++6.1 和 Microsoft Visual C++ 2.0。C++提出了一些更为深入的概念,它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性,掌握起来有一定难度。

但是,C是C++的基础,C++语言和C语言在很多方面是兼容的。因此,掌握了C语言,再进一步学习C++就能以一种熟悉的语法来学习面向对象的语言,从而达到事半功倍的目的。

1.2.2 简单 C 程序介绍

下面先通过几个示例,了解C程序的基本结构,进而认识C程序的结构特点。

【例 1.1】编写C程序,在屏幕上输出一串文字。

程序代码如下:

```
#include<stdio.h>
void main()
{
    printf(" This is a C program! ");
}
```

该程序的功能是在屏幕上输出一串文字:

This is a C program!

这应该是最简单的C程序了。该程序的主体是main()函数,其函数体由一对花括弧{}括起来,表示main()函数要做的事情,用语句来表示。

而函数体内的printf()也是一个函数,它的作用就是输出结果。在其他高级语言中,有专门的输出语句完成输出功能,而在C语言中,就要调用类似的函数来输出结果,其实它还是一个C的标准库函数,因此在程序调用它时,就要在程序开始处加上一行“#include<stdio.h>”,关于此问题后面还会详细介绍,而在printf()函数后面加一个分号(;)后,就表示这是一个C语言的语句了。

【例 1.2】编写C程序,求两个整数之和。

我们可以用下面程序(代码)实现:

```
#include<stdio.h>
void main()
{
    printf(" %d ",6+8);
}
```

亦可以用下面程序(代码)实现:

```
#include<stdio.h>
void main()
{
    int a,b,c;
    a=6;
    b=8;
    c=a+b;
    printf("%d ",c);
}
```

这两段程序的功能都是在屏幕上输出:14(整数6与8之和)。

与例1.1比较,上面第二段程序中,main()函数体内多了四行语句,其中“int a,b,c;”是声明

语句, 声明(定义) a 、 b 和 c 为整型(int)变量, 其他新增的三行为赋值语句, 它们分别表示将整数 6 赋予 a , 将整数 8 赋予 b , 将 $a+b$ 的值(也是整数)赋予 c 。

另外我们还会看到 `printf()` 函数的括号()内与例 1.1 也有所不同: “%d” 是输出项的“格式字符串”, 它表示以十进制整数形式输出 c 的值, 关于“格式字符串”将在第 2 章详细介绍。

【例 1.3】 编写 C 程序, 求任意两个整数之中较大数。

程序代码如下:

```
#include<stdio.h>
void main()                /*主函数*/
{
    int max(int x,int y);    /*对被调用函数max的声明*/
    int a,b,c;              /*定义变量a、b、c为整型变量*/
    scanf("%d,%d",&a,&b);    /*为变量a和b输入值*/
    c=max(a,b);             /*调用max()函数,将函数max()的值赋予c*/
    printf("%d ",c);        /*输出c的值*/
}
int max(int x,int y)        /*函数max()首部。定义max()函数,函数值为整型,
                             形式参数x、y为整型*/
{
    int z;                  /*定义函数max()中用到的变量z为整型变量*/
    if(x>y) z=x;
    else z=y;
    return(z);              /*将变量z的值返回*/
}
```

与例 1.1 和例 1.2 比较, 该程序变得复杂了些。本程序由两个函数构成: `main()` 函数和 `max()` 函数, 而函数 `main()` 调用了函数 `max()`。为了说明各语句的作用, 我们在语句每一行的右面加了注释, 它是由“/*”和“*/”括起来的部分, 各行的功能作用已经在注释行中注明, 以后还会详细介绍。值得注意的是 `main()` 函数中 `scanf()` 函数的作用是为 a 和 b 输入值, 它和前面提到的 `printf()` 都是 C 的标准库函数, 但是要注意它们括号内的书写格式是不同的。其实, 例 1.3 可以改写成更简单的如下形式:

【例 1.4】 求任意两个整数之中较大数。

```
#include<stdio.h>
void main()
{
    int a,b,c;              /*定义变量a、b、c为整型变量*/
    scanf(" %d,%d ",&a,&b); /*为变量a和b输入值*/
    if(a>b) c=a;            /*根据为a和b输入的值,判断后将它们中较大者存入c中*/
    else c=b;
    printf(" %d ",c);       /*输出c的值*/
}
```

例 1.3 和例 1.4 两段程序的功能都是在程序运行后, 再从键盘为 a 和 b 输入数据, 如输入 8、6, 按【Enter】键后输出 14。

请读者自己比较例 1.3 和例 1.4 的区别。

1.2.3 C 程序结构特点

通过以上四个例子, 可以看到 C 程序的结构有如下特点:

① 一个 C 程序由一个或多个函数构成。每个 C 程序至少有一个且仅有一个 `main()` 函数构成, 如例 1.1、例 1.2、例 1.4 中都只有一个 `main()` 函数, 或者可以有一个 `main()` 函数和若干个其他函数构成, 如例 1.3 由一个 `main()` 函数和一个 `max()` 函数构成。`main()` 函数被称为主函数。

② C 的每一个函数都由函数首部 and 函数体两部分组成。由一对大括弧 `{ }` 扩起来的部分称为函数体, 函数体由若干 C 语句构成; 大括弧 `{ }` 前面的部分称为函数首部, 后面章节会有更详细的说明。

③ 一个 C 程序总是从主函数开始执行的。无论主函数的位置在哪里, 都是从主函数开始执行, 再调用其他函数, 直至程序结束, 如例 1.3 中可以将主函数和 `max()` 函数的位置互换, 不影响程序的执行。

④ C 语言中没有专门的输入/输出语句。C 程序的输入/输出通过调用系统库函数 `scanf()` 和 `printf()` 来完成, 除此以外 C 标准库提供了大量库函数 (其实每一个库函数就是一段完成特定功能的程序) 供用户使用。调用系统库函数需要在源程序的最前面增加预处理命令 `include` 行。

⑤ 可以用 `/*.....*/` 对 C 程序的任何部分做注释。注释只是为了增加程序的可读性。

1.3 C 语言基本语法元素

1.3.1 C 语言符号集

就像人类语言文字的组成一样, 字符是组成语言的最基本的元素, 计算机语言亦是如此。C 语言和其他计算机语言一样, 它的基本字符集是由英文字母、阿拉伯数字、空格、标点和特殊字符组成。其具体包括:

① 英文字母: 大小写各 26 个, 共 52 个。

② 阿拉伯数字: 0~9 共 10 个。

③ 空格符: 只在字符常量和字符串常量中起作用。在其他地方出现时, 只起间隔作用, 编译程序对它们忽略。因此, 在程序中使用空格符与否, 对程序的编译不发生影响, 但在程序中适当的地方使用空格符将增加程序的清晰性和可读性。

④ 标点和特殊字符: 主要指运算符, 运算符通常由以下 1~2 个特殊符号组成。

+	-	*	/	%	++	--
<	>	=	>=	<=	==	!=
!		&&	^	~		&
<<	>>	()	[]	{}	\	"
?	:	.	,	;		

1.3.2 C 语言中的词汇

在 C 语言中使用的词汇分为六类, 分别是标识符、关键字、运算符、分隔符、常量、注释符。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外,

其余都由用户自定义。C 语言规定，标识符只能是字母(A~Z, a~z)、数字(0~9)、下画线(_)组成的字符串，并且其第一个字符必须是字母或下画线。

如合法的标识符：a、x、3x、BOOK_1、sum5。

非法的标识符：3s、s*T、-3x、bowy-1。

在使用标识符时还必须注意以下几点：

① 标准 C 不限制标识符的长度，但它受各种版本的 C 语言编译系统限制，同时也受到具体计算机的限制。例如，在某版本 C 中规定标识符前 8 位有效，当两个标识符前 8 位相同时，则被认为是同一个标识符。

② 在标识符中，大小写是有区别的。例如，BOOK 和 book 是两个不同的标识符。

③ 标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便阅读理解，做到“顾名思义”。

2. 关键字

关键字是由 C 语言规定的具有特定意义的小写字符串（标识符），通常也称为保留字。用户定义的标识符不应与关键字相同，也不能作为变量或函数名使用，用户只能根据系统的规定来使用它们。根据 ANSI 标准，C 语言可以使用的关键字有 32 个，如表 1-1 所示。

表 1-1 关键字

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	go to	if
int	long	register	short	signed	sizeof	static	return
suct	switch	typedef	union	unsigned	void	volatile	while

3. 运算符

C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成，如前面已经用到过的运算符有“+”（加法运算符）、“>”（大于运算符）和“=”（赋值运算符）等。关于运算符的介绍详见 2.2 节。

4. 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中，分隔各个变量。空格多用于语句各单词之间，用做间隔符。在关键字、标识符之间必须要有一个以上的空格符做间隔，否则将会出现语法错误，例如把 int a;写成 inta;，C 编译器会把 inta 当成一个标识符处理，其结果必然出错。

5. 常量

C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种，如前面已经用到过的常量有 6 和 8（数字常量）、" This is a C program!"（字符串常量）等。更详细的内容在第 2 章中将专门介绍。

6. 注释符

C 语言的注释符是以“/*”开头，并以“*/”结尾的串。在“/*”和“*/”之间的即为注释。程序编译时，不对注释做任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使翻译跳过不做处理，待调试结束后再去掉注释符。