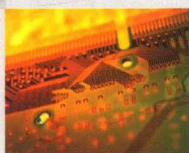


普通高等学校计算机类“十三五”规划教材

C语言程序设计教程

第二版

编 著 韩立毛 徐秀芳



普通高等学校计算机类“十三五”规划教材

C语言程序设计教程

第二版

编著 韩立毛 徐秀芳

内容简介

本书采用任务驱动方式进行教学,以程序设计实例为主导,将知识点融入实例,以实例带动知识点的学习,在按实例教学时,充分注意保证知识的完整性和系统性,通过实例掌握 C 语言程序的设计方法和操作技巧。

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。全书由 11 章组成。主要内容包括:第 1 章 C 语言概述,第 2 章 C 语言程序设计基础,第 3 章顺序结构程序设计,第 4 章选择结构程序设计,第 5 章循环结构程序设计,第 6 章数组与字符串,第 7 章函数与模块化程序设计,第 8 章编译预处理及程序调试,第 9 章指针及其应用,第 10 章自定义数据类型,第 11 章文件及其应用。

本书内容丰富、结构清晰,易于教师进行教学与读者自学。通过 120 个实例的分析讲解,由浅至深,层层引导,使读者能够快速地掌握 C 语言,提高程序设计能力。程序实例有详细的讲解和注释,容易读懂、便于教学。本书在内容选取、概念引入和文字叙述等各方面,都力求遵循面向实际应用、重视实践、便于自学的原则,注重理论联系实际,强调对分析问题、解决问题能力的训练与培养。

本书适合作为高等学校各专业的 C 语言程序设计课程的教材,也可作为从事计算机相关工作的科技人员、广大计算机爱好者的自学读物。

图书在版编目(CIP)数据

C 语言程序设计教程 / 韩立毛, 徐秀芳编著. —2 版.

— 南京:南京大学出版社, 2017. 6

普通高等学校计算机类“十三五”规划教材

ISBN 978 - 7 - 305 - 18952 - 4

I. ①C… II. ①韩… ②徐… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 144494 号

出版发行 南京大学出版社

社 址 南京市汉口路 22 号

邮编 210093

出版人 金鑫荣

丛书名 普通高等学校计算机类“十三五”规划教材

书 名 C 语言程序设计教程

编 著 韩立毛 徐秀芳

责任编辑 何永国

编辑热线 025-83597482

照 排 南京理工大学资产经营有限公司

印 刷 扬州市江扬印务有限公司

开 本 787×1092 1/16 印张 20.5 字数 524 千

版 次 2017 年 6 月第 2 版 2017 年 6 月第 1 次印刷

ISBN 978 - 7 - 305 - 18952 - 4

定 价 39.00 元

网 址: <http://www.njupco.com>

官方微博: <http://weibo.com/njupco>

官方微信号: njupress

销售咨询热线: (025)83594756

* 版权所有, 侵权必究

* 凡购买南大版图书, 如有印装质量问题, 请与所购
图书销售部门联系调换

前 言

C语言从最初的为编写UNIX操作系统而设计,并在实验室内部使用的程序设计语言,发展到独立于UNIX操作系统,并走出实验室为众多的人所关注的、各种计算机上可移植的C语言,再发展到现在普遍采用的标准C语言,使C语言逐渐走向通用化和标准化。由于C语言的简洁、表达能力强、使用灵活方便、运算符和数据结构丰富、生成的代码质量高以及可移植性好等优点,使得C语言倍受人们的青睐,成为结构化程序设计语言中的佼佼者。借助于C语言,人们开发出了很多大型的系统软件和应用软件。

C语言是一种非常出色的程序设计语言,它精练、灵活、应用领域广泛。虽然已经经历了40多个春秋,但至今依然在计算机教学和计算机应用程序设计中起着极其重要的作用。因此,许多高校将C语言程序设计列为大学生的必修基础课程。为了使读者能够在学习C语言程序设计的过程中始终保持强烈的学习兴趣,领悟到程序设计的奥妙,掌握并使用C语言程序设计解决相关专业的实际问题,我们结合多年的教学实践和丰富的教学经验编写了这本《C语言程序设计教程》。

全书共11章。第1章C语言概述。从介绍C语言的发展和特点入手,并结合了一些实例展示C语言的概貌,同时简单地介绍了C语言程序的开发过程;第2章C语言程序设计基础。介绍了程序设计的概念、C语言常量、简单变量以及基本数据类型、运算符和表达式等基础知识,同时还介绍了运算符的优先级和结合性;第3章顺序结构程序设计。介绍了数据的输入和输出、计算和处理,顺序结构程序设计的方法和步骤;第4章选择结构程序设计。介绍了选择结构语句以及选择结构程序设计的方法和步骤;第5章循环结构程序设计。介绍了当型循环、直到型循环、计数型循环、循环嵌套以及循环结构程序设计的方法和步骤;第6章数组与字符串。介绍了数组的基本概念,数组的定义和使用方法,还介绍了字符串及其处理技术;第7章函数与模块化程序设计。介绍了函数的定义与调用以及函数的嵌套和递归调用,还介绍了变量的作用域与存储类别;第8章编译预处理及程序调试。介绍了编译预处理的宏定义、文件包含和条件编译,程序调试的基本方法与技巧;第9章指针及其应用。介绍了指针的概念和使用方法,指针与数组以及指针的应用;第10章自定义数据类型。介绍了结构体与共用体的概念、结构体数组以及结构体指针,同时还介绍了枚举类型;第11章文件及其应用。介绍了数据文件的基本概念、文件的操作方法以及文件的应用。附录中给出了ASCII代码表、C语言关键字与运算符和C语言库函数。

本书覆盖了C语言的主要语言点,对C语言的介绍比较全面和系统。为了使读者能够真正掌握C语言,在介绍C语言的各个语言点时,力求做到通俗易懂,尤其是对语言点中容易出现错误的地方作了详细的说明,并结合了应用实例,使读者能够真正加以运用。本书内容丰

富、结构清晰,易于教师进行教学与读者自学。书中具有较大的知识信息量,从程序设计的基础知识、程序流程控制、数组、指针、自定义数据类型到最后的文件操作,通过 120 个实例的分析讲解,利用 280 道习题进行练习与巩固,再通过 48 道上机练习题的操作实践,由浅至深,层层引导,使读者能够快速地掌握 C 语言,提高程序设计能力。程序实例均有详细的讲解和注释,容易读懂、便于教学。本书在内容选取、概念引入和文字叙述等各方面,都力求遵循面向实际应用、重视实践、便于自学的原则,注重理论联系实际,强调对分析问题、解决问题能力的训练与培养。

全书采用任务驱动方式进行讲解,以程序实例为主导,将知识点融入实例,以实例带动知识点的学习。在按实例进行讲解时,充分注意保证知识的相对完整性和系统性,使读者通过学习实例掌握 C 语言的程序设计方法和程序设计技巧。

本书由韩立毛和徐秀芳共同编写,其中第 1、2、3、8、9 章由徐秀芳编写,第 4、5、6、7、10、11 章由韩立毛编写,并由韩立毛对全书进行了统稿。本书在编写过程中南京大学徐宝文教授给予了许多指导,得到了江苏省应用型本科高校计算机系列教材编委会的支持,得到了盐城工学院教材出版基金的资助,同时也得到了我校赵雪梅、董琴、张成彬等老师和南京大学出版社的热心帮助和大力支持,在此一并表示感谢。

由于作者水平有限,书中难免存在错误和不足之处,恳请读者批评指正。

编著者

2017 年 6 月修订

目 录

第 1 章 C 语言概述	1
1.1 C 语言的引入	1
1.2 C 语言程序的基本结构	5
1.3 C 语言程序的上机过程与步骤	8
1.4 C 语言的简单应用实例	11
练习题 1	12
第 2 章 C 语言程序设计基础	15
2.1 程序设计的引入	15
2.2 算法与流程图	16
2.3 标识符、常量与变量	22
2.4 基本数据类型	24
2.5 运算符和表达式	31
2.6 类型转换	40
2.7 常用库函数	41
2.8 C 语言程序设计基础应用实例	45
练习题 2	47
第 3 章 顺序结构程序设计	51
3.1 顺序结构程序的引入	51
3.2 数据的输入	52
3.3 数据的输出	57
3.4 基本语句及程序规范	61
3.5 顺序结构程序设计及实例	62
练习题 3	65
第 4 章 选择结构程序设计	69
4.1 选择结构程序的引入	69
4.2 条件语句	70
4.3 开关语句	80

4.4 选择结构程序设计及实例	83
练习题 4	91
第 5 章 循环结构程序设计	97
5.1 循环结构程序的引入	97
5.2 当型循环	98
5.3 直到型循环	101
5.4 计数型循环	103
5.5 循环嵌套与辅助控制	108
5.6 循环结构程序设计及实例	114
练习题 5	119
第 6 章 数组与字符串	125
6.1 数组与字符串的引入	125
6.2 一维数组	126
6.3 二维数组	130
6.4 字符串及其处理	134
6.5 数组与字符串程序设计及实例	139
练习题 6	150
第 7 章 函数与模块化程序设计	158
7.1 函数与模块化程序设计的引入	158
7.2 函数的定义	159
7.3 函数的调用和返回值	160
7.4 函数的嵌套调用与递归调用	164
7.5 数组作为函数参数	170
7.6 变量的作用域与存储类别	173
7.7 函数与模块化程序设计及实例	179
练习题 7	185
第 8 章 编译预处理及程序调试	193
8.1 编译预处理及程序调试的引入	193
8.2 宏定义	194
8.3 文件包含	199
8.4 条件编译	201

8.5 程序调试	202
8.6 综合实例	206
练习题 8	209

第 9 章 指针及其应用 212

9.1 指针的引入	212
9.2 指针与指针变量	213
9.3 指针与数组	224
9.4 指针应用及实例	247
练习题 9	251

第 10 章 自定义数据类型 255

10.1 自定义数据类型的引入	255
10.2 结构体类型	255
10.3 共用体类型	268
10.4 枚举类型	271
10.5 类型定义符	275
10.6 自定义数据类型程序设计及实例	277
练习题 10	285

第 11 章 文件及其应用 290

11.1 文件的引入	290
11.2 文件的打开与关闭	291
11.3 文件的顺序读(写)	293
11.4 文件的随机读(写)	301
11.5 文件应用程序设计及实例	303
练习题 11	304

附录 A ASCII 代码表 308

A.1 标准 ASCII 代码	308
A.2 扩展 ASCII 代码	309

附录 B C 语言关键字与运算符 310

B.1 C 语言关键字及功能	310
B.2 C 语言运算符的优先级与结合性	311

附录 C C 语言库函数	312
C.1 数学函数	312
C.2 字符函数	313
C.3 字符串函数	314
C.4 输入/输出函数	315
C.5 动态存储分配函数	317
C.6 其他函数	317
参考文献	319

第1章 C语言概述

本章的主要内容:计算机语言的概念,C语言程序的基本结构,C语言程序的上机过程与步骤。通过本章内容的学习,了解C语言的起源、发展和特点,理解C语言程序的基本结构,熟悉C语言程序的开发过程,初步学会使用Visual C++6.0的集成开发环境,掌握使用C语言程序解决实际问题的基本方法。

1.1 C语言的引入

【任务】 利用计算机解决简单的实际问题,如计算地球赤道的长度和地球的表面积等,掌握使用C语言程序设计解决问题的具体过程。

1.1.1 程序和程序设计的概念

你想掌握如何使用计算机解决上述实际问题吗?你想知道计算机是如何自动进行工作的吗?要回答这两个问题,你就必须学习并掌握程序和程序设计方面的知识。

你要知道,计算机是不能自己直接解决问题的,而是要你(程序设计人员)先编好程序,然后将程序输入计算机,接下来再运行程序。计算机是根据人们事先编制好的程序自动进行工作的,而程序则是通过程序设计,用程序设计语言编制的。所谓程序就是为实现特定目标或解决特定问题用计算机语言编写的命令序列集合,而程序设计则是按指定要求编写计算机能够识别的特定指令组合的过程。要使用计算机解决各类问题,我们就应该学会用程序设计语言进行程序设计。

1.1.2 程序设计语言

语言是交流的工具,自然语言是人与人之间交流的工具,而程序设计语言则是人与计算机之间的交流工具。程序设计语言又称编程语言,用程序设计语言编写的程序能被计算机系统所接受、理解和执行,且随着计算机技术的发展而不断地发展。

计算机语言按其发展一般分为机器语言、汇编语言和高级语言,又可以分为低级语言和高级语言两大类,低级语言包括机器语言和汇编语言。低级语言直接依赖计算机硬件,不同的机型所用的低级语言不同;高级语言不再依赖计算机硬件,用高级语言编写的程序可以方便地、几乎不加修改地用在不同类型的计算机上。

1. 机器语言

机器语言是第一代程序设计语言,是直接由二进制代码指令表达的计算机语言,是由“0”和“1”组成的代码串。例如,一个字长为16位的计算机,它是由16个二进制数字组成一条指令或其他信息。机器语言能被计算机直接识别和执行,不需要进行任何翻译,具有灵活、直接执行和速度快等特点。

每种类型计算机的机器指令格式和代码所代表的含义都是硬件规定的。当把在某台计算机上执行的程序移植到另一台计算机上执行时,必须重新编写程序,因此机器语言可移植性

差,重用性差。另外,由于机器语言是由“0”和“1”组成,编程人员首先要熟记计算机的全部指令和代码含义才能进行程序编写,因此用机器语言编写程序是一项极其烦琐的工作。

2. 汇编语言

为克服机器语言中“0”和“1”给程序员带来的不便,汇编语言用助记符代替操作码,用地址符或标号代替地址码。如算术运算指令中的 ADD 表示加法,SUB 表示减法,MUL 表示乘法,DIV 表示除法等。

使用汇编语言编写的程序,计算机不能直接识别,需要将汇编语言程序翻译成机器语言程序,这种起翻译作用的程序称为汇编程序,把汇编语言程序翻译成机器语言程序的过程称为汇编。汇编语言作为面向机器的低级语言,保持了机器语言的优点,具有直接和简洁的特点,可有效访问、控制计算机的各种硬件设备,目标代码短,占用内存少,执行速度快。尽管如此,相对高级语言来说,汇编语言在编写复杂程序时代码量较大,又由于它与处理器密切相关,而每种处理器都有自己的指令系统,相应的汇编语言也各不相同,所以,汇编语言程序的通用性,可移植性较差。

3. 高级语言

为增加程序的可读性,提高程序的可维护性和可移植性,可以采用一种比较接近自然语言和数学语言的程序设计语言,即高级语言来实现。使用高级语言编写的程序不能直接运行,需要经过“翻译”,将高级语言程序翻译成机器语言程序才能被计算机运行。

高级语言程序的“翻译”通常有解释方式和编译方式两种。解释方式是对高级语言编写的程序翻译一句执行一句;而编译方式是将高级语言编写的程序文件全部翻译成机器语言,生成可执行文件后再执行。

在计算机的发展史上,先后出现过几百种高级语言,其中,影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、Pascal、C、PROLOG、Ada、C++、Visual C++、Visual BASIC、Java 等。

高级语言的出现大大提高了程序员的工作效率,降低了程序设计的难度,并改善了程序的质量,其良好的可读性和可维护性,也使更多的人掌握了程序设计方法,从而使计算机技术得到快速发展和普及。

4. 三类语言的比较

三类语言的比较如表 1-1 所示。

表 1-1 三类语言的比较

编程语言	描述
机器语言	1. 计算机能够直接识别并执行,由“0”和“1”代码串组成。 2. 程序员非常难于记忆和识别。
汇编语言	1. 可以直接对硬件操作,由指令、伪操作和宏指令组成。 2. 程序用英文缩写的标识符,程序员更加容易识别和记忆。 3. 程序经汇编生成的可执行文件小、执行速度快。
高级语言	1. 屏蔽编程语言的相关细节,如寄存器、堆栈编程。 2. 程序需要通过“翻译”后才能被计算机执行。 3. 学习较容易,有良好的可读性和可维护性。

1.1.3 C语言的发展

1. C语言的由来

C语言的原型是ALGOL 60语言,也称为A语言。1963年,剑桥大学将之发展为CPL语言(Combined Programming Language),并进一步简化为BCPL语言。1970年,美国贝尔实验室的Ken Thompson将BCPL进行了修改,并取名为“B语言”,两年后,D. M. Ritchie在B语言的基础上最终设计出了一种新的语言,取BCPL的第二个字母“C”,这就是C语言。1978年美国电话电报公司(AT&T)贝尔实验室正式发布了C语言。

2. C语言的标准化

1978年,由Brian W. Kernighan和Dennis M. Ritchie合著了著名的《The C programming language》一书,通常简称为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,后来美国国家标准化协会(ANSI)在此基础上为C语言制定了一套ANSI标准,于1983年发表,通常称之为ANSI C。在1989年进行修订,成为现行的C语言标准(1999年发布C99标准),随后出现了遵循该标准的各种C和C++集成开发环境。在学习C语言的过程中,我们将会体验C语言的魅力以及编程带来的乐趣。

3. C语言的应用领域

C语言的应用领域非常广泛,如人工智能、单片机控制、计算机系统软件等的底层开发、工业控制、智能仪表、嵌入式系统、硬件驱动程序开发等。

近年来,在每年的“大学生机器人比赛”、“全国大学生飞思卡尔杯智能车竞赛”以及某些国际、国内大型企业举办或赞助举办的面向在校大学生的各类电子竞技中,许多参赛院校都是使用C语言来编制其控制程序的。因此有“学好C语言,走遍天下都不怕”的说法。另外,学好C语言也为学习其他面向对象的程序设计语言(如C++、Java等)奠定了基础。

4. 基于C语言的语言及特点

20世纪80年代中期以后,面向对象程序设计语言、面向对象程序设计方法广泛应用于程序设计,全新的、面向对象的程序设计语言被开发出来。同时对传统语言全面进行面向对象的扩展,C++语言是其代表,这类语言既支持传统的面向过程的程序设计,又支持新型的、面向对象的程序设计。下面以C++、Java、C#为例,介绍其与C语言的联系。

(1) C++语言。C++是C语言的超集,C++对C语言的最大改进是引进了面向对象机制,同时C++依然支持所有C语言特性,保留对C语言的兼容。C++适合于开发桌面程序和游戏后台开发。

(2) Java语言。Java是基于C++的,也继承了C语言的许多特性。Java是一种可以编写跨平台应用程序的面向对象的程序设计语言,是由Sun公司于1995年5月推出的Java程序设计和Java平台(即JavaSE, JavaEE, JavaME)的总称。Java技术具有卓越的通用性、高效性、平台移植性和安全性,广泛应用于个人PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业社群。

(3) C#语言。C#(C Sharp)是由微软公司开发的一种面向对象的、运行于.NET Framework之上的高级程序设计语言。由C++和Java发展而来,侧重于网络与数据库编程。

注意:Microsoft Visual C++(简称Visual C++、MSVC、VC++或VC)是微软公司的C++开发工具,具有集成开发环境,可提供编辑C语言、C++等编程语言。

目前,在微机上广泛使用的 C 语言编译系统有 Microsoft C(MSC)、Turbo C(TC)、Borland C(BC)、Visual C++ 6.0 的集成开发环境等。这些版本不仅实现了 ANSI C 标准,而且在此基础上各自作了一些扩充,有其自身的特点,更加方便、完美。本书选定的上机环境是 Visual C++6.0。

1.1.4 C 语言的特点

C 语言经过不断的发展和完善,成为当今计算机界公认的一种优秀程序设计语言,有着其他语言不可比拟的特点。具体来说,C 语言的主要特点如下:

1. 适合开发系统软件

C 语言最初是为了编写 UNIX 操作系统而开发,它既具有高级语言的易学、易用、可移植性强的特点,又具有低级语言执行效率高,可对硬件进行操作等优点;既适合开发系统软件,也可以开发应用软件。

2. 结构化的程序设计语言

C 语言是一种结构化语言,它提供了编写结构化的基本控制语句,并以具有独立功能的函数作为模块化程序设计的基本单位,有利于以模块化方式进行设计、编码、调试和维护。

3. 具有丰富的数据类型和表达式

C 语言不仅本身提供了大量的数据类型,如整型、实型、字符型等,还可以由用户根据自己设计需要定义特殊的数据类型,同时还允许大多数数据类型之间进行转换。运算符和表达式的类型丰富多样,包括赋值运算符、条件运算符、算术运算符、逗号运算符等 30 多种。

4. 可移植性好

由于 C 语言程序本身并不依赖于机器硬件,且 UNIX、WINDOWS、DOS 等主要的操作系统都支持 C 语言编译器,因此,C 语言程序基本不做任何修改就能运行于各种类型的计算机和操作系统环境上。

5. 语句简洁、结构紧凑、功能强

C 语言中只提供了 32 个关键字,9 条控制语句。编程风格灵活,语法限制少,易于阅读和维护。

6. 具有预处理功能和丰富的库函数

预处理的使用为程序的修改、阅读、移植和调试提供了方便。同样,大量的库函数可供程序设计人员直接调用,省去了重复编写这些函数的时间和精力,大大提高了程序设计的效率,并保证了程序设计的质量。

7. 面向对象程序设计的基础

1980 年,Bjarne Stroustrup 开发了 C++。C 语言中几乎所有的功能和特点都被 C++ 所吸收,并在此基础上对 C 语言进行了全面的改进,增加了大量的新特性(其中最重要的特性就是“类”)。C++ 是 C 语言改进后的产物,现已经被广泛用于各种软件的开发,同时仍在不断地完善中。经过改进和补充,C++ 已发展成为面向对象程序设计语言的代表。

当然,C 语言也有其自身的弱点,如语法限制不太严格,在增加程序设计灵活性的同时,在一定程度上降低了某些安全性,这也对程序设计人员提出更高的要求。

C 语言是一种面向过程的、灵活的结构化程序设计语言。在计算机日益普及的今天,C 语言的应用领域依然非常广泛,几乎各类计算机都支持 C 语言的开发环境。而 C 语言也是学习 C++、Java 等其他语言的基础。

1.2 C语言程序的基本结构

在了解C语言程序之前,我们先通过几个实例对C语言程序的结构有一个初步认识。本节主要介绍C语言程序的基本结构。

1.2.1 简单C语言程序实例

【例 1.1】 利用C语言程序计算两个整数之和。

(1) 算法分析。

这是一个简单的求和运算,由于程序和数据要放到内存才能被执行,因此,需要3个元素的存储空间才能存放加数、被加数与和。此处定义a,b两个变量并赋值,表示加数和被加数,定义sum存放a与b的和,然后输出两者之和的结果。

(2) 程序设计。

```
1  / * ----- 例 1.1.c ----- */
2  #include <stdio.h> /* 编译预处理命令 */
3  void main() /* 定义主函数 */
4  { /* 函数开始标志 */
5      int a,b,sum; /* 声明a,b,sum为整型变量 */
6      a=1; b=2; /* 给变量a,b赋初值 */
7      sum=a+b; /* 计算a+b,并将结果放在变量sum中 */
8      printf("sum=%d\n",sum); /* 输出结果 */
9  } /* 函数结束标志 */
```

(3) 程序运行。

程序运行结果如下:

```
sum=3
```

(4) 程序说明。

第1行表示注释部分,为便于理解,可用汉字表示注释,当然也可以用英语或汉语拼音作注释。注释对编译和运行不起作用,只是为了增加程序的可读性,便于维护。注释可以加在程序中任何位置。

第2行是以#开始的编译预处理命令,是在编译系统翻译代码前需由预处理程序处理的语句。本例中#include <stdio.h>语句是请求预处理程序将文件stdio.h包含到程序中,作为程序的一部分。

第3行中,main是函数名,函数名后面必须跟一对圆括号,括号内可以定义形参,也可以没有。main()前面的void表示函数值的类型是void类型(空值类型)。

第4行和第9行分别表示函数的开始和结束标志。

第5行是变量声明部分,将变量a,b和sum定义为整型(int)变量。

第6行是两个赋值语句,使变量a和b的值分别为1和2。

第7行求和,使sum的值为a+b。

第8行为输出语句,使用标准输出函数printf,以指定格式输出变量sum的值。“%d”表示以十进制整数形式输出sum的值,printf函数中括弧内最右端sum是要输出的变量,现在它的值为3(即1+2的值),因此输出一行信息为sum=3。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(详见第3章)。

【例 1.2】 利用 C 语言程序求两个数中的较大值。

(1) 算法分析。

求两个数较大值,可以事先定义两个变量并赋初值后,将两个数进行比较,并输出较大的数。也可以将求两个数较大值的过程用自定义函数来实现,可以供不同的函数调用,更具有通用性。本例以调用自定义函数的方式实现求两个数较大值,介绍 C 语言中函数的主要用法。

(2) 程序设计。

```
/* -----例 1.2.c----- */
#include <stdio.h> /* 编译预处理命令 */
void main() /* 告诉编译器,C 程序由此开始执行 */
{ /* 程序执行开始 */
    int max(int x,int y); /* 对被调用函数 max 的声明 */
    int a,b,c; /* 声明 a,b,c 为整型变量 */
    printf("请输入 a,b 的值:"); /* 在屏幕上显示“请输入 a,b 的值:” */
    scanf("%d,%d",&a,&b); /* 由键盘输入 a,b 的值 */
    c=max(a,b); /* 调用 max 的函数值,并赋值给 c */
    printf("较大值是:%d",c); /* 输出 a,b 比较后的结果 */
} /* 程序执行结束 */

/* 以下是用户自定义函数,求两个整数中的较大值的 max 函数 */
int max(int x,int y) /* 定义 max 函数,函数值为整型 */
{
    int z; /* 声明 z 为整型变量 */
    if(x>y)z=x; /* 如果 x>y 成立,则将 x 的值赋给 z */
    else z=y; /* 如果 x>y 不成立,则将 y 的值赋给 z */
    return z; /* 将 z 的值返回给函数 max */
}
```

(3) 程序运行。

程序运行结果如下:

① 第一次运行结果。

请输入 a,b 的值:15,25

较大值是:25

② 第二次运行结果。

请输入 a,b 的值:8,-5

较大值是:8

(4) 程序说明。

程序代码中的注释较为详细地说明了每行代码的功能,具体的函数定义及调用方式详见第7章。

例 1.1 所示的 C 语言程序仅由一个 main() 函数构成,它相当于其他高级语言中的主程序;例 1.2 所示的是由一个 main() 函数和一个其他函数 max() (自己设计的函数) 构成,函数 max() 相当于其他高级语言的子程序。

1.2.2 C语言程序的基本结构

1. 程序的组成

一个完整的C语言程序是由一个main()函数(又称主函数)和若干个其他函数结合而成的,或仅由一个main()函数构成。每个函数完成一定的功能,函数参数是被函数处理的数据,函数参数能够在函数与函数之间传递数据。

每个完整的C程序都必须有且仅有一个main函数,程序总是从main函数开始执行,而main函数可以位于源程序文件中的任何位置。main函数是程序执行的入口,其他函数的执行是由main函数中的语句调用来完成的。被调函数既可以是由系统提供的库函数,也可以是由设计人员自己根据需要而设计的函数。

例如,在例1.2中,printf()函数是C语言编译系统库函数中的一个函数,它的作用是在屏幕上按指定格式输出指定的内容;而max()函数则是由用户自己设计的函数,它的作用是求两个数中的较大值。

2. 函数的结构

函数是一个独立的程序块可以相互调用,但不能相互嵌套。并且main函数以外的任何函数只能由main函数或其他函数调用,自己不能单独运行。

一个函数由函数首部 and 函数体两部分组成,其一般格式如下:

[函数类型] 函数名([函数形式参数表])

{

 数据说明部分;

 函数执行部分;

}

(1) 函数首部,即函数的第一行。包括函数返回值类型、函数名、函数属性、形式参数类型、形式参数名。例如,例1.2中的max函数的首部为int max(int x,int y),分别表示函数返回值为整型,函数名为max,有两个分别定义的整型参x和y。

一个函数名后面必须跟一对圆括号,括号内写函数的参数类型及参数名。如果函数没有参数,可以在括号中写void,也可以是空括号,如main(void)或main()。

(2) 函数体,即函数首部下面的大括弧{...}内的部分。如果一个函数内有多对大括弧,则最外层的一对{}为函数体的范围。

函数体一般包括:

① 数据说明部分。由变量定义、自定义函数声明和外部变量说明等部分组成,其中变量定义是主要的。

例如,例1.2中main函数中的“int a,b,c;”。

② 函数执行部分。函数执行部分一般由若干条可执行语句组成。

例如,在例1.1的main()函数中,除变量定义语句“int a,b,sum;”外,其余4条语句构成该函数可执行的语句部分。

当然,在某些情况下也可以没有声明部分,甚至可以既无声明部分,也无执行部分。

3. 程序的基本特点

(1) 函数体中的数据说明部分必须位于可执行语句之前,即数据说明语句不能与可执行语句交叉在一起。

例如,下面程序中变量定义语句“int max;”的语句位置是错误的。

```
void main()
```

```
{
```

```
int x,y;
```

```
x=3; y=7;
```

```
int max;
```

/* 此句位置错误,所有的定义都应在可执行语句之前 */

```
if(x>y)max=x;
```

```
else max=y;
```

```
printf("max=%d\n",max);
```

```
}
```

如何改正,请读者思考。

(2) C 程序书写格式自由,一行内可以写一个或多个语句,一个语句也可以分写在多行上。

(3) C 语言本身没有输入/输出语句。输入和输出的操作是由库函数 scanf 和 printf 等函数来完成的,C 语言对输入/输出实行“函数化”。

(4) 可以用“/* */”对 C 程序中的任何部分作注释(在 VC++ 中也可用“//”作为行注释,本章中采用第一种方法进行注释,后面各章中均采用第二种方法进行注释)。一个好的、有使用价值的源程序都应当加上必要的注释,以增加程序的可读性。

(5) C 语言中严格区分字母的大小写。一般使用小写字母作为函数名、变量名等,而使用大写字母作为常量名。

(6) C 语言规定每条语句或数据说明均以分号“;”结束,否则编译时可能会出错。

(7) 主函数 main()既可以放在 max()函数之前,也可以放在 max()函数之后。习惯上,将主函数 main()放在最前面。

(8) 为增强可读性,最好以缩进的格式书写程序(若不遵守,也不影响程序运行)。

(9) 使用“{ }”时,为便于检查匹配性,最好同一层次的“{”和“}”缩进相同(若不遵守,也不影响程序运行)。

1.3 C 语言程序的上机过程与步骤

1.3.1 C 语言程序的上机过程

编写一个 C 语言程序的上机过程一般为:编辑→编译→连接→执行。

(1) 编辑。使用一个文本编辑器编辑 C 语言源程序,并将其保存为文件扩展名为“.c”的文件。

(2) 编译。将编辑好的 C 语言源程序翻译成二进制目标代码的过程。编译过程由 C 语言编译系统自动完成。编译时首先检查源程序的每一条语句是否有语法错误,当发现错误时,就在屏幕上显示错误的位置和错误类型信息,此时要再次调用编辑器进行查错并修改,然后再进行编译,直到排除所有的语言和语义错误。正确的源程序文件经过编译后,在磁盘上生成同名的目标文件(.obj)。

(3) 连接。将目标文件和库函数等连接在一起形成一个扩展名为“.exe”的可执行文件。如果函数名称写错或漏写包含库函数的头文件,则可能提示错误信息,从而得到程序错误