



普通高等教育“十三五”应用型人才培养规划教材

C语言程序设计

C Language Programming

◎ 主编 祁文青 刘志远 冯运仿



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十三五”应用型人才培养规划教材

C 语言程序设计

主 编 祁文青 刘志远 冯运仿

副主编 姚 莉 曹绍君 臧 辉

参 编 吕 璐 谢 晋

机械工业出版社

根据 TIOBE 编程语言排行榜,近十年,C 语言一直位于使用热门率前两位,这是由其自身的特性所决定的。它既具备高级语言特点,又可实现对底层硬件的操控;既能编写系统软件,也能编写应用软件。因此,在各高校的入门语言中,C 语言一直是被选择最多的语言。

本书较系统地介绍了 C 语言的基本概念、程序设计的基本方法和技巧,结构合理、思路清晰、语言简练。同时,本书从培养学生程序开发能力入手,在每章均配有由浅入深的案例和习题,可增强学生对基本概念的理解和解决实际问题的能力。

为了方便广大师生的教学和学习,本书还提供了配套的多媒体课件、例题和习题源代码等教学资源。

本书适合作为高等学校各专业的 C 语言程序设计课程教材,也可作为从事计算机相关工作的科技人员、计算机爱好者及各类自学人员的参考书。

本书配有电子课件,欢迎选用本书作教材的老师登录 www.cmpedu.com 注册下载,或发 [jinacmp@163.com](mailto:jnacmp@163.com) 索取。

图书在版编目 (CIP) 数据

C 语言程序设计/祁文青,刘志远,冯运仿主编. —北京:
机械工业出版社,2018.8

普通高等教育“十三五”应用型人才培养规划教材
ISBN 978-7-111-60389-4

I. ①C… II. ①祁… ②刘… ③冯… 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 (2018) 第 147389 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:吉玲 责任编辑:吉玲

责任校对:张薇 封面设计:张静

责任印制:孙炜

天津翔远印刷有限公司印刷

2018 年 8 月第 1 版第 1 次印刷

184mm × 260mm · 19.25 印张 · 470 千字

标准书号:ISBN 978-7-111-60389-4

定价:45.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:010-88379833

机工官网:www.cmpbook.com

读者购书热线:010-88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面无防伪标均为盗版

金书网:www.golden-book.com

前言

C 语言的功能强大,使用灵活方便,移植性强,兼有高级语言和低级语言的特点,利用 C 语言可以编写系统软件和应用软件。正因如此,C 语言一直是各高校计算机专业首选的入门语言,同时也是其他专业重要的公共基础课程之一。

作者根据多年的教学经验,分析了国内外多种同类教材的优缺点,在此基础上,编写了本书。全书内容丰富、结构合理、语言简练。全书共 11 章,内容包括:程序设计与 C 语言、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、指针、函数、字符串、用户自定义数据类型、动态内存分配、文件。

本书在关注重心上做了大胆改革,传统教材主要介绍 C 语言的概念和语法,而本书的重心则是现实环境中的各类数据对象,并通过恰当的方法来存储数据、解决问题,这种改革,使得学生不再是停留在概念记忆和语法理解上,而是通过理论联系实际,引导和启发学生掌握思考和解决问题的方法,达到举一反三的目的。

本书在每一章都安排了类型丰富的案例和习题,并已在 Visual C++6.0 环境下调试运行通过。题目难度由浅入深,教师可根据学生实际水平选择部分习题在课堂完成,加强学生对概念的掌握,也可布置其他综合性题目,来培养学生解决实际问题的能力。循序渐进地启发学生逐步提高程序设计能力,强调程序的多种解法和优化、引导学生在“游泳中学会游泳”。

本书在后几章中对指针、函数、字符串、用户自定义数据类型、动态内存分配、文件等内容的介绍既全面具体、又简洁明了,为后续数据结构、操作系统、组成原理等课程的学习打下了良好的基础。

为了方便广大师生的教学和学习,本书还提供了配套的多媒体课件、例题和习题源代码等教学资源。

本书由湖北理工学院计算机学院组织编写,祁文青、刘志远、冯运仿任主编,姚莉、曹绍君、藏辉任副主编,吕璐、谢晋参编。全书由祁文青负责统稿。

本书可作为本科、高职高专教材或参考书,同时也可供广大自学者作为自修课本。

由于编者水平有限,书中难免存在不当和疏漏之处,恳请读者原谅,并提出宝贵意见。

编者

目 录 Contents

前 言	
第 1 章 程序设计与 C 语言	1
1.1 程序设计	1
1.2 编程语言	2
1.3 C 语言简介	4
1.4 简单的 C 程序介绍	5
1.5 运行 C 程序的步骤与方法	8
1.6 C 语言常用词	8
1.7 本章知识点小结	10
1.8 本章常见错误小结	10
习题	11
第 2 章 顺序结构程序设计	12
2.1 算法与程序	12
2.2 常量和变量	13
2.2.1 常量	13
2.2.2 变量	16
2.3 数据类型	17
2.3.1 数据类型的分类	17
2.3.2 数据类型所占内存空间的大小	18
2.3.3 运算符和表达式	21
2.3.4 数据类型之间的转换	25
2.4 C 语句	26
2.4.1 C 语句的作用和分类	26
2.4.2 赋值语句	27
2.5 数据的输入/输出	28
2.5.1 字符数据的输入/输出	28
2.5.2 格式输入/输出	29
2.6 顺序结构程序应用举例	33
2.7 本章知识点小结	36
2.8 本章常见错误小结	37
习题	37
第 3 章 选择结构程序设计	39
3.1 用 if 语句实现选择结构	39
3.2 关系运算符和关系表达式	43
3.3 逻辑运算符和逻辑表达式	46
3.4 条件运算符和条件表达式	49
3.5 用 switch 语句实现多分支 选择结构	50
3.6 选择结构程序应用举例	51
3.7 本章知识点小结	54
3.8 本章常见错误小结	55
习题	56
第 4 章 循环结构程序设计	58
4.1 用 while 语句实现循环	58
4.2 用 do-while 语句实现循环	62
4.3 用 for 语句实现循环	64
4.4 改变循环执行的状态	66
4.5 循环的嵌套	69
4.6 循环程序应用举例	71
4.7 本章知识点小结	73
4.8 本章常见错误小结	74
习题	75
第 5 章 数组	76
5.1 一维数组的定义和初始化	76
5.2 一维数组元素的输入/输出	78
5.3 一维数组应用举例	81
5.4 二维数组的定义和初始化	88
5.5 二维数组元素的输入/输出	91
5.6 二维数组应用举例	92
5.7 本章知识点小结	93
5.8 本章常见错误小结	94
习题	94
第 6 章 指针	97
6.1 指针的基本概念	97
6.2 指针变量的定义及使用	98
6.3 指针和一维数组间的关系	102
6.4 指针和二维数组间的关系	107
6.5 指针数组	111
6.6 指向指针的指针变量	112

6.7	基本数据类型、数组类型、指针 数据类型的比较	113	习题	198
6.8	本章知识点小结	117	第9章 用户自定义数据类型	200
6.9	本章常见错误小结	118	9.1 结构体类型	200
习题	119	9.1.1 结构体类型的定义	200
第7章 函数	122	9.1.2 结构体变量及结构体指针 变量的定义	202	
7.1 函数的基本概念	122	9.1.3 结构体变量的使用及初始化	203	
7.2 函数定义	123	9.1.4 结构体变量作函数参数	206	
7.3 函数调用	126	9.1.5 结构体指针变量作函数参数	208	
7.3.1 函数的形式参数和实际参数	126	9.1.6 结构体数组的定义和初始化	210	
7.3.2 函数返回值	129	9.1.7 结构体数组作函数参数	212	
7.3.3 函数原型	129	9.1.8 结构体程序应用举例	213	
7.4 函数形式参数的类型	132	9.2 共用体类型	219	
7.4.1 基本类型变量作函数 形式参数	132	9.2.1 共用体类型的定义	219	
7.4.2 指针变量作函数形式参数	134	9.2.2 共用体类型变量的定义 及初始化	220	
7.4.3 一维数组作函数形式参数	137	9.2.3 共用体程序应用举例	222	
7.4.4 二维数组作函数形式参数	141	9.3 枚举类型	224	
7.5 函数的嵌套调用	143	9.3.1 枚举类型的定义	224	
7.6 函数的递归调用	145	9.3.2 枚举类型变量的定义	225	
7.7 指向函数的指针	147	9.3.3 枚举类型程序应用举例	226	
7.8 变量的作用域和存储属性	152	9.4 类型定义符 typedef	227	
7.8.1 变量的作用域	152	9.5 本章知识点小结	229	
7.8.2 变量的存储属性	155	9.6 本章常见错误小结	230	
7.8.3 关于函数和变量的声明和定义	160	习题	232	
7.9 返回指针值的函数	160	第10章 动态内存分配	233	
7.10 模块化程序设计	163	10.1 动态内存分配的基本概念	233	
7.11 本章知识点扩充内容	164	10.2 动态内存分配系统函数	234	
7.12 本章知识点小结	166	10.3 动态数组——数据的顺序存储	238	
7.13 本章常见错误小结	167	10.4 单向链表——数据的链式存储	246	
习题	169	10.4.1 链式存储的基本概念	246	
第8章 字符串	173	10.4.2 单向链表的基本操作	247	
8.1 字符串的基本概念	173	10.5 本章知识点小结	257	
8.2 字符串的存储及输入/输出	174	10.6 本章常见错误小结	258	
8.2.1 用字符数组存储字符串 及输入/输出	174	习题	259	
8.2.2 用字符指针存储字符串 及输入/输出	177	第11章 文件	261	
8.3 字符串处理函数	179	11.1 文件的基本概念	261	
8.4 字符串应用举例	182	11.1.1 文本文件及二进制文件	261	
8.5 自定义字符串处理函数	188	11.1.2 文件缓冲区	262	
8.6 本章知识点小结	197	11.1.3 FILE 指针	263	
8.7 本章常见错误小结	198	11.1.4 文件位置指针	264	
		11.2 文件的基本操作	264	

11.3 文件的读/写操作	268	习题	287
11.3.1 字符读/写函数	268	附录	289
11.3.2 字符串读/写函数	271	附录 A C语言中32个关键字详解	289
11.3.3 格式化读/写函数	274	附录 B C运算符的优先级与结合性	290
11.3.4 数据块读/写函数	279	附录 C 常用字符与ASCII码值对照表	292
11.4 本章知识点小结	286	附录 D 常用的ANSI C标准库函数	293
11.5 本章常见错误小结	287	参考文献	300

第 1 章

程序设计与 C 语言

1945 年 6 月，冯·诺依曼提出了在数字计算机内部的存储器中存放程序的概念，这是所有现代电子计算机的模板，称为“冯·诺依曼结构”，按这一结构建造的计算机称为存储程序计算机，又称为通用计算机。冯·诺依曼计算机主要由运算器、控制器、存储器和输入/输出设备组成。它的特点是：程序以二进制代码的形式存放在存储器中；所有的指令都是由操作码和地址码组成的；指令按照其执行的顺序进行存储；以运算器和控制器作为计算机结构的中心等。

计算机指令是指能被计算机识别和执行的命令。计算机程序就是计算机为完成某一任务所必须执行的一系列指令的集合。通过这些指令集合，计算机可以实现数值计算、非数值（文本、图形、图像等）计算等功能。

计算机语言是人和计算机进行交流所使用的语言，用计算机语言即程序设计语言来编写计算机程序的过程就叫程序设计。

1.1 程序设计

计算机的一切操作都是由程序控制的，离开程序，计算机将一事无成。程序设计是指设计、编制、调试程序的方法和过程，是目标明确的智力活动。程序设计通常分为问题建模、算法设计、编写代码、编译调试和编写程序文档五个阶段。

1. 问题建模

对于接受的任务要进行认真地分析，研究所给定的条件，分析最后应达到的目标，找出解决问题的规律，选择解题的方法，完成实际问题建模。

2. 算法设计

算法设计即设计出解题的方法和具体步骤。

3. 编写代码

将算法翻译成计算机程序设计语言，对源程序进行编辑、编译和链接。

4. 编译调试

运行可执行程序，得到运行结果。能得到运行结果并不意味着程序正确，要对结果进行分析，看它是否合理。若不合理，要对程序进行调试，即通过上机发现和排除程序中的故障。

5. 编写程序文档

许多程序是提供给别人使用的，如同正式的产品应当提供产品说明书一样。正式提供给用户使用的程序，必须向用户提供程序说明书，内容应包括程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据以及使用注意事项等。

1.2 编程语言

计算机不理解人类的语言，所以，计算机程序必须使用计算机能够识别和执行的语言编写。计算机程序设计语言经历了从机器语言、汇编语言到高级语言的发展历程。

1. 机器语言

1946年2月14日，世界上第一台通用计算机ENIAC诞生于美国的宾夕法尼亚大学，这台机器上使用了一种特定的穿孔卡片，卡片使用0和1组成的语言（相当于硬件的通电状态为1，无电状态为0）操作计算机进行工作，这种语言与人类的自然语言差别极大，只有专业的编程人员才能理解，称为机器语言。机器语言是第一个计算机语言，后续编程语言都是在其基础上发展变化而来的。机器语言的本质是一组使用二进制代码表示的机器指令的集合，能够被计算机识别并执行。但是，不同型号的机器语言无法通用，使用机器语言编写的程序全是0、1代码，既不容易被使用者理解，也容易出错，修改起来十分繁琐。

2. 汇编语言

用机器语言进行程序设计是非常单调乏味的过程，通过开发人员不懈的努力，创建了汇编语言，计算机语言进入第二阶段。汇编语言是一种用于计算机或其他电子设备的编程语言，它用助记符代替了操作码，用标号或地址符号代替了地址码，所以也称为符号语言。在不同的机器中，汇编语言对应着不同的机器语言指令集。汇编语言保留了机器语言高速度和高效率、便于记忆和书写的特点，降低了程序设计的难度。汇编语言仍是面向机器的语言，设计出来的程序不易被移植，所以不像其他大多数的高级计算机语言一样被广泛应用。用汇编语言写代码需要知道CPU是如何工作的，它通常应用于底层程序优化或硬件操作。

3. 高级语言

20世纪50年代，新一代编程语言——高级语言出现了，计算机语言步入到第三阶段。高级语言容易识记和理解，是大多数编程者的首选编程语言。与机器语言相反，高级语言以人类的日常用语为基础，是一种接近于人类语言习惯的编程语言。它使用人们易于接受的文字（如英文），程序中的符号也与日常用的数学公式有关，这大大提高了程序的可读性。高级语言还具有远离机器语言的特点，消除了环境特异性带来的代码移植的困难，利用率高。高级语言作为编程语言发展的重要里程碑，自动化程度高，表达形式多样且灵活，将繁琐的事务抛给了编译程序，可以说是对程序员的一次解放。

(1) FORTRAN 语言

FORTRAN语言是由美国著名的计算机先驱人物约翰·巴克斯（John W. Backus）于1954年提出的。FORTRAN是FORmula TRANslator的缩写，意思是“公式翻译机”。顾名思义，该语言主要用于科学计算。FORTRAN自推出之日起，版本不断更新，功能不断增强，目前在工程应用领域，FORTRAN仍然被广泛使用。

(2) COBOL 语言

COBOL（COmmon Business Oriented Language，通用事务处理语言）是在美国国防部推动下，由政府机构和工业界联合开发的一种语言，于1960年正式推出，主要用于商业数据处理。COBOL语言曾经使用非常广泛，20世纪70年代近一半的程序是用COBOL语言编写的。当前在商业领域，COBOL语言仍然占有重要席位。

(3) BASIC 语言

BASIC (Beginner's All-purpose Symbolic Instruction Code, 初学者的通用符号指令代码) 是1964年由美国的 John G. Kemeny 和 Thomas E. Kurtz 在 FORTRAN 语言的基础上开发的。由于简单易学, BASIC 语言得到了广泛普及。Microsoft 公司对 BASIC 可谓是一往情深, 从早期微型机上内置的 BASIC, 到20世纪80年代产生的第一个编译版本 Quick BASIC, 直到目前非常流行的 Visual Basic, 一直没有中断过对 BASIC 语言的改进。最新出现的 Visual Basic .NET, 是采用 Microsoft 的 .NET 技术的 Visual Basic 语言。

(4) Pascal 语言

Pascal 是由瑞士计算机科学家 Niklaus Wirth 设计的一种语言, 1968年提出后被全世界广泛接受。这个语言的名字是为了纪念著名的法国数学家, 也是计算科学的先驱 Blaise Pascal 而起的。由于结构小巧、语法严谨、数据类型丰富, 从20世纪70年代末往后的很长一段时间里, Pascal 成为世界范围的计算机专业教学语言。20世纪80年代, 随着C语言的流行, Pascal 走向了衰落。目前, 在商业上仅有 Borland 公司仍在开发基于 Pascal 语言系统的 Delphi。它使用了面向对象与软件组件的概念, 主要用于开发商用软件。

(5) C 与 C++ 语言

C 语言是由美国贝尔实验室的 Kennet L. Thompson 和 Dennis M. Ritchie 于1972年设计开发的, 当时主要用于编写 UNIX 操作系统。后来由于其功能丰富、使用灵活、执行速度快、可移植性强, 迅速成为最广泛使用的程序设计语言之一。C 语言既可以用来开发系统软件, 也可以用来开发应用软件, 应用领域很广泛。例如, 在中国广泛使用的计算机辅助设计软件 AutoCAD、数学软件系统 Mathematica 等, 以及许多语言编译系统本身, 其软件系统的全部或部分都是用 C 语言开发的。C 语言已经成为最重要的软件系统开发语言之一。

1980年, 贝尔实验室的 Bjarne Stroustrup 对 C 语言进行了扩充, 加入了面向对象的概念, 并于1983年改名为 C++。目前, C++ 已经成为应用最广的面向对象程序设计语言。Microsoft 公司的 Visual C++ 和 Borland 公司的 C++ Builder 是 C++ 语言最常用的开发工具。利用这些开发工具可以高效率地开发出复杂的 Windows 应用程序。最新出现的 C# 语言使用了 C++ 的语法和语义, 是基于 Microsoft 公司推出的新一代软件开发环境 .NET 平台的高级程序设计语言。

(6) Java 语言

Java 是 Sun 公司开发的一种跨平台的网络编程语言, 于1995年正式发布。其语言风格与 C++ 接近, 但舍弃了 C++ 中一些不常用或容易被误用的成分, 如指针等。Java 语言最主要的特点是, 同一个 Java 程序不用重新编译就可以在不同平台的计算机上运行。Java 在网络上的独特优势以及其跨平台的特点, 使得它已经成为 Internet 上最受欢迎的编程语言之一。

(7) 网页设计类语言

目前, 最常用的三种动态网页设计语言是 ASP (Active Server Pages)、JSP (Java Server Pages)、PHP (Hypertext Preprocessor)。三者都提供在 HTML 代码中混合某种程序代码、由语言引擎解释执行程序代码的能力。ASP 是一个 Web 服务器端的开发环境, 利用它可以产生和执行动态的、互动的、高性能的 Web 服务应用程序。ASP 支持 VBScript、JScript 等脚本语言。JSP 是用 Java 作为脚本语言的, 并可以在 Servlet 和 JavaBean 的支持下, 完成功能强

大的站点程序。PHP 是一种跨平台的服务器端的嵌入式脚本语言，大量地借用 C、Java 和 Perl 语言的语法，并融合自身的特性，使 Web 开发者能够快速写出动态生成页面。

高级语言的表示形式近似于自然语言，对各种公式的表示近似于数学公式，而且一条高级语言语句的功能往往相当于十几条甚至几十条汇编语言的指令，程序编写相对简单。因此，在工程计算、数据处理、信息处理等方面，人们常用高级语言来编写程序。

计算机无法识别高级语言编写的程序，需要一种称为编译程序的软件把高级语言写的程序（也称为源程序或源代码）转换为机器语言的代码（称为目标程序），计算机才能执行。

4. 非过程化的程序语言

人们称高级语言是第 3 代语言，其特点是面向过程。面向过程是指用户在程序中不但要说明解决什么问题，还要告诉计算机如何去解决。

计算机技术的发展要求新一代的计算机语言能够根据用户说明的问题，智能化地去自动寻找解决方案，具有这种功能的语言称为第 4 代语言。目前第 4 代语言尚未发展成熟，主要面向基于数据库应用的领域，还不适用于科学计算、高速实时系统和系统软件等的开发。

1.3 C 语言简介

1978 年，布莱恩·柯林汉（Brian Kernighan）和丹尼斯·里奇（Dennis Ritchie）制作了 C 的第一个公开可用的描述，现在称为 K&R 标准。

UNIX 操作系统、C 编译器和几乎所有的 UNIX 应用程序都是用 C 语言编写的，C 语言现在已经成为一种广泛使用的专业语言。C 对现代编程语言有着很大的影响，许多优秀的语言都借鉴于 C 语言。C 语言标准是于 1988 年由美国国家标准协会（American National Standard Institute, ANSI）制定的。当今最流行的 Linux 操作系统和关系数据库管理系统（Relational Database Management System, RDBMS）MySQL 都是使用 C 语言编写的。

C 语言作为一种计算机程序设计语言，既具有高级语言的特点，又具有汇编语言的特点。它可以作为工作系统设计语言，编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。它的应用范围广泛，具备很强的数据处理能力，不仅仅适用于系统软件开发，也适用于应用系统开发。C 语言具有以下特点：

1. 简洁紧凑、灵活方便

C 语言只有 32 个关键字，9 种控制语句，程序书写形式自由，区分大小写，且把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作，这三者是计算机最基本的工作单元。

2. 运算符丰富

C 语言的运算符包含的范围很广泛，共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 语言的运算类型极其丰富，表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

3. 数据类型丰富

C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂的数据结构的运算，并引入了指针概念，使程序效率更高。另外，C 语言具有强大的图形功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

4. 结构式语言

结构式语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C语言是以函数形式提供给用户的,这些函数可被方便地调用,并具有多种循环、条件语句控制程序流向,从而使程序完全结构化。

5. 语法限制不太严格,程序设计自由度高

虽然C语言也是强类型语言,但它的语法比较灵活,允许程序编写者有较大的自由度。由于C语言允许直接访问物理地址,可以直接对硬件进行操作,因此它既具有高级语言的功能,又具有低级语言的许多功能,能够像汇编语言一样对位、字节和地址进行操作,可用于编写系统软件。

6. 生成目标代码质量高,程序执行效率高

C语言一般只比汇编程序生成的目标代码效率低10%~20%。

7. 适用范围大,可移植性好

C语言有一个突出的优点就是适合于多种操作系统,如DOS、UNIX、Windows98、Windows NT等,也适用于多种机型。C语言具有强大的绘图能力,可移植性好,并具备很强的数据处理能力,是适于数值计算的高级语言。

1.4 简单的C程序介绍

为了说明C语言源程序结构的特点,先看以下几个程序,可从这些例子中了解组成一个C源程序的基本部分和书写格式。

【例 1.1】 在屏幕上输出“Hello, C world!”。

```
#include <stdio.h>
int main()
{
    printf("Hello,C world!\n");
    return 0;
}
```

运行结果:

```
Hello,C world!
```

程序分析:

1) 如何让计算机开口说话,计算机开口说话就是将要说的话显示在显示器屏幕上。本程序的任务是让计算机跟人问好。

2) 此程序的第一行称为C语言的编译预处理命令,每一个需要键盘输入数据或屏幕输出数据的程序都需要使用这条命令。stdio.h是扩展名为.h的头文件,h为head之意,std为standard之意,i为input之意,o为output之意。编译预处理命令#include可以使头文件在程序中生效,头文件stdio.h中包含实现数据输入和输出的C函数。

3) 每一个C源程序都必须有且只能有一个main函数。

4) return 0 表示 main 函数已经被正常地执行完毕, 这个 0 是返回给 main 函数的调用者的, 用来告知程序执行的结果是否正常。

5) printf() 函数的功能是把要输出的内容送到显示器去显示, 将需要显示的内容放括号里面, 并且使用双引号引起。\\n 表示光标移动到下一行。

6) C 语言中用分号表示一个语句结束。

通过上面的学习, 了解到如何让计算机“说话”, 接下来看看, 如何让计算机做数学运算。首先回想一下启蒙之初扳着手指学加法的情景: 小红有 3 本书, 小明又送给她 1 本, 小红一共有几本书? 仔细分析一下, 得出结果的过程大致分为 5 个步骤:

- 1) 伸出左手 3 个手指, 代表小红已有书本数量, 记录左手的书本数量;
- 2) 伸出右手 1 个手指, 代表小明送出的书本数量, 记录右手的书本数量;
- 3) 将左右手的手指数量相加;
- 4) 获得结果;
- 5) 将结果说出来。

如果要让计算机做加法, 同样需要以上几个步骤。但在此之前, 还要解决一个问题, 如何让计算机像人类的大脑一样记住数字的值? 这里需要使用计算机的存储空间来记录程序执行过程中的数据, 用下面的代码可以完成对上述问题的计算。

【例 1.2】 小红有 3 本书, 小明又送给她 1 本, 小红一共有几本书?

```
#include <stdio.h>
main()
{
int a,b,c;      //定义了 a,b,c 三个整型变量
a=3;           //用 a 存储小红的书本数量
b=1;           //用 b 存储小明送出的书本数量
c=a+b;         //用 c 存储书本数量的总和
printf("小红一共有%d本书。\\n",c);
return 0;
}
```

运行结果:

小红一共有 4 本书。

程序分析:

1) 例 1.1 中未使用任何变量, 因此无说明部分。C 语言规定, 源程序中所有用到的变量都必须先定义, 后使用, 否则将会出错。本例中定义了 3 个变量, a、b 分别用来存储两个整数, 变量 c 存储 a 与 b 的和, 调用 printf 函数在显示器上输出 c 的值。

2) “=” 为赋值运算符, 将对应的整数或表达式的值存入 a、b、c 中。虽然在书写上和数学中的等号相同, 但两者的含义在本质上是不同的, 赋值运算符的含义是将右侧整数或表达式的值赋给左侧的变量。注意, C 语言中赋值运算符的操作是有方向性的, 并无“等号两侧操作数的值相等”之意。

3) “%d” 为格式符号, 表示按十进制整数的形式输出变量的值, 不能写成“printf (“小

红一共有c本书。”);”，这样写显示器上只会显示“小红一共有c本书。”。

4) 以上程序语句依次逐行执行，称为顺序结构，和后续学习的选择结构、循环结构并称为C语言的三种基本结构。

5) C语言注释方法有两种：

多行注释：/* 注释内容 */

单行注释：//注释一行

注释是写给程序员看的，C语言编译器在编译时，不对注释做任何处理。注释可出现在程序中的任何位置，用来向用户提示或解释程序的意义。

【例 1.3】 小红有 10.5 元钱，借给小明 2.7 元，小红还剩多少钱？

```
#include <stdio.h>
float subtract(float x, float y)
{
    float z;
    z = x - y;
    return (z);
}
main()
{
    float a, b;
    a = 10.5;
    b = 2.7;
    a = subtract(a, b);
    printf("小红还剩%f 元钱\n", a);
    return 0;
}
```

运行结果：

小红还剩 7.800000 元钱

程序分析：float 表示定义 a 和 b 为实数型变量，可以存储实数；%f 表示以小数形式输出变量的值。

通过以上三个程序总结 C 源程序的结构特点如下：

1) 一个 C 语言源程序可以由一个或多个源文件组成。

2) 每个源文件可由一个或多个函数组成。C 语言称为函数式语言，C 程序的工作都是由各式各样的函数完成的，函数可分为库函数和用户定义函数两种。库函数是由 C 系统提供的，用户无需定义，也不必在程序中做类型说明，只需在程序前包含有该函数原型的头文件即可在程序中直接调用，如程序用到的 printf 函数。用户定义函数是由用户按需要定义的函数，如 subtract 函数。所有自定义的函数都必须遵循“先定义，后使用”的原则。

3) 一个源程序不论由多少个文件组成，都有一个且只能有一个 main 函数，即主函数。

4) 源程序中可以有预处理命令（include 命令仅为其中的一种），预处理命令通常应放

在源文件或源程序的最前面。

5) 每一个说明、每一条语句都必须以分号结尾,但预处理命令、函数头和花括号之后不能加分号。

6) 标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,也可不再加空格来间隔。

从书写清晰,便于阅读、理解、维护的角度出发,在书写 C 程序时应遵循以下规则:

1) 一个说明或一条语句占一行。

2) 用 {} 括起来的部分通常表示程序的某一层结构。{} 一般与该结构语句的第一个字母对齐,并单独占一行。

3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,以便看起来更加清晰,增加程序的可读性。

在编程时应力求遵循这些规则,以养成良好的编程风格。通过这三个简单的 C 程序可以看出,C 程序主要包括预处理指令、主函数、定义变量、对变量进行赋值运算和算术运算、输出运算结果、注释等。

1.5 运行 C 程序的步骤与方法

C 语言本身没有数据输入/输出对应的指令,所有的数据输入/输出都是由库函数完成的。程序员编程时,只需要简单地调用这些标准函数即可,当编译工具把程序员编写的源程序编译成机器语言时,遇到其中的 printf 函数,并不能转换成机器语言。这样编译的程序称为目标程序,以 .obj 为后缀。为了产生真正可以运行的程序,还需要将编译好的目标程序与编程语言提供的库文件中某些函数的指令连接在一起。这个步骤称为链接(Link),只有经过链接的程序才能产生可执行的 .exe 文件。

C 语言的编程步骤如下:

1) 上机输入和编辑源程序 (.c 文件);

2) 对源程序进行编译 (.obj 文件);

3) 进行链接处理 (.exe 文件);

4) 运行可执行程序,得到运行结果。

需要说明的是,不同语言编译的方式不同。有的语言是先将所有程序代码一起编译成机器语言,再链接生成可执行文件,如 C 语言、Pascal 语言,这种语言称为编译型语言,最后以可执行的 .exe 文件运行;有的语言则可以边编译边执行,如 BASIC 语言、Java 语言,这种语言称为解释型语言;也有些语言既提供编译运行的方式,也提供解释运行的方式,如 Visual Basic,在调试程序时可以采用解释型,一旦调试完成,则采用编译型,将源程序编译成可执行的 .exe 文件。编译型语言的程序执行速度比解释型语言的程序执行速度快。

1.6 C 语言常见用词

在 C 语言中常见的用词有标识符、关键字、运算符、分隔符、常量、注释符等。

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)、下划线(_)组成的字符串,并且其第一个字符必须是字母或下划线。

以下标识符是合法的:

a、x、x3、BOOK_1、sum5。

以下标识符是非法的:

3s 以数字开头;

s*T 出现非法字符*;

-3x 以减号开头;

bowy-1 出现非法字符-(减号)。

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,同时也受到具体机器的限制。例如,在某版本C中规定标识符前八位有效,当两个标识符前八位相同时,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,以便于阅读理解,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字符串,通常也称为保留字。用户定义的标识符不应与关键字相同。C语言的关键字分为以下几类:

1) 类型说明符:用于定义或说明变量、函数或其他数据结构的类型,如前面例题中用到的int、double等。

2) 语句定义符:用于表示一个语句的功能,如return就是返回语句的语句定义符。

3) 预处理命令字:用于表示一个预处理命令,如前面各例中用到的include。

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。

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做间隔符。在关键字、标识符之间必须要有一个及以上的空格做间隔符,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。

6. 注释符

C语言的注释有多行注释和单行注释,多行注释是以“/*”开头并以“*/”结尾的串,单行注释是以“//”开始的一行。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。

1.7 本章知识点小结

内 容	概 述	备 注
关键字	关键字就是已被 C 语言编辑工具本身使用, 不能作为其他用途使用的字	C 语言只有 32 个关键字
控制语句	控制语句即用来对程序流程的选择、循环、转向和返回等进行控制	C 语言有 9 种控制语句
主函数	C 语言中主函数指的是 main 函数, 它是所有程序运行的入口	每个源程序有且只有一个主函数, 系统总是从该函数开始执行 C 语言程序
字符集	C 语言基本字符集分为源字符集 (书写 C 语言源文件所用的字符集) 和执行字符集 (C 语言程序执行期间解释的字符集)。源字符集包括字母 (52 个)、数字 (10 个)、格式符 (4 个)、特殊字符 (29 个)。执行字符集在源字符集的基础上还包括空字符、行末标志符 (换行符)、警报符、退格符 (BS) 和回车符 (CR)	C 语言程序中允许出现的字符集是 ASCII 码
注释	注释就是被编译器忽略, 仅供修改阅读的说明	注释可以出现在程序中的任何地方
算法	算法 (Algorithm) 是指解题方案的准确而完整的描述, 是一系列解决问题的清晰指令	

1.8 本章常见错误小结

常见错误举例	常见错误描述	错误类型
<pre>#incdlue <stdio.h> intmian() { printf("HELLO!"); }</pre>	include, main 和 printf 等关键字的拼写错误	编译错误
<pre>#include <stdio.h> int main() { printf("HELLO!"); }</pre>	程序中所有标点为全角符号	编译错误
<pre>#include <stdio.h> int main() { printf("HELLO!") }</pre>	程序中每句结束时要有分号	编译错误