

全国高等学校计算机公共课程优秀教材
中国大学MOOC（慕课）课程主讲教材



C Programming Language
Fourth Edition

C语言程序设计

(第4版)

黄洪艺 李慧琪 张丽丽◎编著

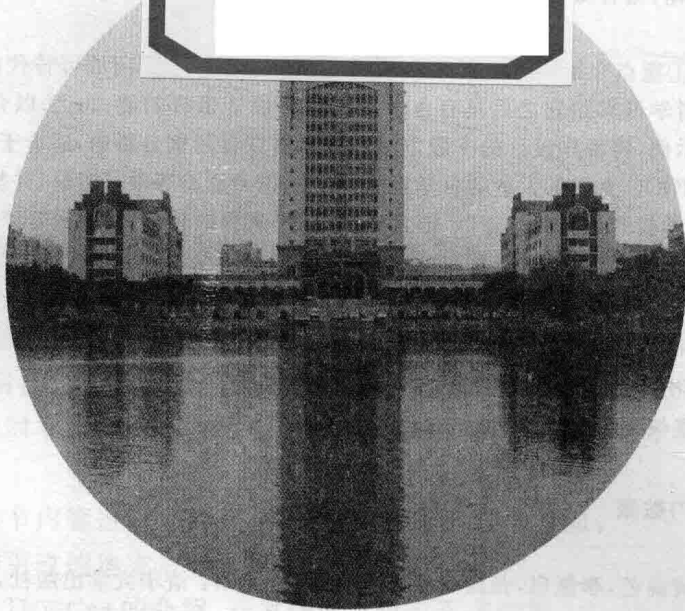
Huang Hongyi

Li Huiqi

Zhang Lili



清华大学出版社



C Programming Language
Fourth Edition

C语言程序设计

(第4版)

黄洪艺 李慧琪 张丽丽◎编著
Huang Hongyi Li Huiqi Zhang Lili

清华大学出版社
北京

内 容 简 介

本书内容包括程序设计语言和程序设计的基本概念,C 语言的词法语法,三种基本控制结构,函数的定义和调用,数组、结构体、指针等复杂数据类型的定义和应用,文件操作等。并且每章都配有例题和练习题。

本书的特点如下:①重点介绍 C 语言基本的、常用的语法,忽略不常用或可替代的语句;②注重程序设计语言的共性,使读者学习 C 语言之后具有自学其他程序设计语言的能力;③以介绍 C 语言的语法为线索,通过精心组织的示例,将程序设计的一般方法和技术贯穿在示例分析中,适合于案例教学。

本书在中国大学 MOOC 上提供大量的自学视频、课件、单元测验等学习资料,并提供讨论与答疑。

本书适合作为高等学校非计算机专业“C 语言程序设计”课程的教材,特别适合于周授课 3 学时或 3 学时以下的学生使用。由于配套 MOOC 资源,本书也适合作为自学编程者的入门教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计/黄洪艺,李慧琪,张丽丽编著.—4 版.—北京:清华大学出版社,2017
ISBN 978-7-302-48193-5

I. ①C… II. ①黄… ②李… ③张… III. ①C 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 207850 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:时翠兰

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京泽宇印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:15.75 字 数:385 千字

版 次:2006 年 9 月第 1 版 2017 年 9 月第 4 版 印 次:2017 年 9 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:076173-01

前言

PREFACE

“C 语言程序设计”是高校非计算机专业学生的编程入门课程,厦门大学公共计算机教学部已开设该门课程超过 20 年。通过长期的计算机基础教学实践与研究,老师们都积累了丰富的经验,并先后修订了 3 次教材,前 3 版都是基于 ANSI C 的标准编写的。随着 C 标准与时俱进地更新,目前的 C99 标准与 ANSI C 标准已有较多差异,有些与本书涉及的内容相关。因此促使编者对本书再次进行改版,并配套丰富的教辅资料,具体工作内容如下:

1. 修订内容

第 4 版对各章节内容进行了修改与补充,包括以下 5 个方面:

(1) 对 C 标准更改的地方进行了更新。

(2) 增加了对 Dev-C++ 的介绍,开发环境可以有更多选择。

(3) 为第 1~4、6 章增设了问答题,这些章节语法知识点较多,记清有一定的难度。问答题多以选项的形式给出,是对主要知识点的归纳。通过问答,厘清与编程相关的主要或易错知识点。

(4) 从易于理解的角度出发,对部分教材内容进行了补充与修改。

(5) 选择题、编程题也做了一定程度的扩充,使习题更有针对性,更有梯度。

在对教材进行改版的过程中,同时保留了第 3 版的特色,注重案例教学,将编程方法与技巧融入各案例分析中。C 语言是一门实践性很强的课程,学好 C 语言的最佳途径就是编程,多写多练。

2. 配套丰富的教辅资源

为推动信息技术与 C 语言教学的深度融合,进一步提高教学质量,收获更好的教学效果,厦门大学公共计算机教学部启动在线开放课程的建设,并于 2016 年 10 月 15 日在中国大学 MOOC 开课,课程链接为 <http://www.icourse163.org/course/XMU-1001771003>,其中有大量的教辅资源。

教辅资源内容包括以下 4 类:

1) 75 个教学短视频

MOOC 课程与传统上课不同,学生是课程学习的主人,教师是学生学习的引导者和辅助者,从“教”为中心改为以“学”为中心。为激发学生的学习积极性,我们对教学内容进行合理规划,把知识点碎片化为一个个不超过 10 分钟的视频,精心组织每个视频的内容,充分使用现代技术将知识点以易于吸收的形式展示。

视频完全免费观看。每轮开课分 13 周在线发布,与课堂教学同步。学生可以自选时间观看,每轮课程结束后仍然一直开放,参与者可以随时观看。

2) PDF 讲稿

可以免费自由下载。讲稿简洁清晰,以另一种方式阐述教学内容,便于复习与归纳。

3) 在线单元测试与作业

每周知识配有单元测试题,系统自动评分。答题结束后,可以看到每道题的解析。单元测试题以巩固知识点为目的,同时通过自主做题掌握重点、难点,学习编程技巧。作业题是编程题,侧重于应用,也是书后习题的有益补充。

4) 讨论、答疑

重要章节设计有课堂讨论题,是对所学知识的扩充与引导。积极参与课堂讨论,有助于语法的深入理解及学以致用。对于学习过程中的任何问题,也可以在讨论区中提出,同学和老师都可以参与解答,在互助的环境中学习。

到2017年6月10日截止,已完成两轮开课,视频、单元测试等各类资源已较成熟。在此,要诚挚地感谢庄朝晖、曾华琳老师的参与,使得内容更为丰富完整。还要特别地感谢黄保和老师,他一直支持MOOC建设,并为《C语言程序设计(第4版)》的编写无私地提供材料与建议。最后还要感谢厦门大学公共教学部的各位老师,他们为教材的改进一如既往地提供帮助与支持。

《C语言程序设计(第4版)》由黄洪艺策划和统稿。黄洪艺编写第1、2、3、11章,李慧琪编写第4、5、10章,张丽丽编写第6、7、8、9章。

“C语言程序设计”课程是厦门大学重点建设的慕课课程,本课程可到“中国大学MOOC”(http://www.icourse163.org/)观看学习。每年春、秋两季开课,开课期间,在“中国大学MOOC”首页搜索“C程序设计基础”,即可看到课程并进入学习。也可通过上页介绍的课程链接进入课程。

在使用本书过程中,如有宝贵意见和建议,恳请与黄洪艺联系(邮箱:hyhuang@xmu.edu.cn)。

编者

2017年8月

目录

CONTENTS

第 1 章 绪论	1
1.1 程序设计语言	1
1.2 程序设计	2
1.2.1 程序设计概念	2
1.2.2 算法	2
1.2.3 程序设计的步骤	2
1.3 C 语言发展和 C++ 简介	4
1.3.1 C 语言发展简述	4
1.3.2 C++ 简介	5
1.3.3 集成开发环境	5
1.4 C 语言程序的构成	5
1.5 Visual C++ 简介	9
1.5.1 运行简单 C 程序	9
1.5.2 程序调试一般过程和手段	11
1.5.3 Visual C++ 调试方法和工具	12
1.6 Dev-C++ 5.11 简介	14
1.6.1 C 程序的编辑与运行	15
1.6.2 Dev-C++ 调试方法和工具	16
习题	18
第 2 章 C 语言基础	21
2.1 C 语言词法	21
2.1.1 基本字符集	21
2.1.2 关键字	21
2.1.3 特定字	22
2.1.4 标识符	22
2.1.5 运算符	22
2.1.6 分隔符	23
2.2 C 语言的数据类型	23
2.2.1 数据类型概述	23
2.2.2 基本数据类型	24
2.3 常量与变量	26
2.3.1 常量	26
2.3.2 变量	32

2.3.3	常量与变量应用举例	33
2.4	表达式	36
2.4.1	表达式概述	36
2.4.2	算术表达式	37
2.4.3	类型转换	38
2.4.4	赋值表达式	40
2.4.5	自增、自减表达式	42
2.4.6	逗号表达式	43
习题	44
第3章	结构程序设计	48
3.1	结构化程序设计方法	48
3.2	语句的概念	50
3.3	输入输出函数	52
3.3.1	格式输出函数	52
3.3.2	格式输入函数	57
3.3.3	字符输出函数	59
3.3.4	字符输入函数	59
3.4	顺序结构程序设计举例	60
习题	62
第4章	选择结构程序设计	65
4.1	关系表达式和逻辑表达式	65
4.1.1	关系表达式	65
4.1.2	逻辑表达式	66
4.2	if 语句	67
4.2.1	if 语句	67
4.2.2	if...else 语句	70
4.2.3	if 语句的嵌套	72
4.2.4	if...else if 语句	75
4.2.5	条件表达式	78
4.3	switch 语句	79
习题	81
第5章	循环结构程序设计	87
5.1	for 语句	87
5.2	while 语句	93
5.3	do...while 语句	97
5.4	循环的嵌套	100
5.5	break 语句和 continue 语句	102
5.6	goto 语句	104
5.7	常用算法举例	105
习题	110
第6章	函数	116
6.1	函数定义与调用	116

6.1.1	函数定义	118
6.1.2	函数调用	119
6.1.3	函数原型声明	122
6.2	函数间数据传递	123
6.2.1	函数参数	124
6.2.2	函数返回值	125
6.3	函数的嵌套与递归	126
6.3.1	函数嵌套调用	127
6.3.2	函数递归调用	128
6.4	函数应用举例	130
6.5	变量属性	133
6.5.1	变量的生存期和可见性	133
6.5.2	变量的作用域	133
6.5.3	变量的存储类别	136
	习题	139
第7章	编译预处理	144
7.1	宏定义	144
7.1.1	不带参数的宏	144
7.1.2	带参数的宏	146
7.1.3	取消宏定义	148
7.2	文件包含	148
7.3	条件编译	150
7.3.1	#if 和 #endif 命令	150
7.3.2	#ifdef 和 #ifndef 命令	151
7.3.3	defined 预处理运算符	152
	习题	152
第8章	数组	155
8.1	一维数组	155
8.1.1	一维数组的定义	155
8.1.2	一维数组的引用	156
8.1.3	一维数组的初始化	158
8.1.4	一维数组应用举例	159
8.2	多维数组	161
8.2.1	二维数组的定义和引用	161
8.2.2	二维数组的初始化	162
8.2.3	二维数组应用举例	162
8.3	字符串	164
8.3.1	字符型数组	164
8.3.2	字符串	165
8.3.3	字符串处理函数	167
8.3.4	字符串应用举例	170
	习题	171

第 9 章 结构体、共用体和枚举类型	174
9.1 结构体	174
9.1.1 结构体类型的定义	174
9.1.2 结构体变量定义和初始化	175
9.1.3 结构体变量的引用	177
9.1.4 结构体数组	179
9.2 共用体	182
9.2.1 共用体类型的定义	182
9.2.2 共用体变量的定义	183
9.3 枚举类型	185
9.3.1 枚举类型的定义	185
9.3.2 枚举变量的定义	186
9.4 typedef 语句	188
习题	189
第 10 章 指针	191
10.1 地址与指针变量	191
10.1.1 内存单元地址	191
10.1.2 指针	191
10.1.3 指针变量的定义和初始化	192
10.1.4 指针的运算	193
10.2 指针与函数	195
10.2.1 指针变量作为函数参数	195
10.2.2 函数的返回值为指针	197
10.2.3 指向函数的指针	198
10.3 指针与数组	199
10.3.1 一维数组与指针	199
10.3.2 字符串与指针	203
10.3.3 指针数组	205
10.4 指针与结构体	209
10.4.1 指向结构体的指针	209
10.4.2 动态存储分配	211
10.4.3 链表	212
习题	217
第 11 章 文件	221
11.1 文件概述	221
11.2 文件的打开和关闭	223
11.2.1 文件的打开	223
11.2.2 文件的关闭	224
11.3 文件的读写	225
11.3.1 文本文件的读写	225
11.3.2 二进制文件的读写	229

C 语言是当今世界上使用最广泛的程序设计语言之一。本章将简单回顾程序设计语言的发展历程,概述程序设计、算法的概念及程序设计的主要步骤,再介绍 C 语言的产生和发展以及 C 程序的构成、编辑、调试与测试等。

1.1 程序设计语言

用计算机解决问题,就要编写程序。所谓程序,是指以某种程序设计语言为工具对解决问题的操作序列的描述。程序表达了解决问题的过程,用于指挥计算机进行一系列操作,从而实现问题的解决。程序设计语言就是用户用于编写程序的语言,是一种人造语言。程序设计语言是计算机软件系统的重要组成部分,可分为机器语言、汇编语言和高级语言。用高级语言编写的程序(称其为源程序)要通过编译程序翻译成机器语言程序后计算机才能执行,或者通过解释程序对源程序边解释边执行。

1. 机器语言

最初的程序是用机器语言编写的,机器语言由一系列基本指令组成,这些指令可以由机器直接执行。机器语言编写的程序是由二进制代码组成的代码序列。使用机器指令进行程序设计要求程序设计者具有广博的计算机专业知识,对机器的硬件有充分的了解,这种程序的可读性差,而且由于不同机器的指令系统不同,程序的可移植性差,大大地限制了程序的通用性。此外,用机器语言描述问题的处理方式也与人们习惯的思维方式有较大差距。

2. 汇编语言

为了便于记忆,人们很自然地用助记符号来代表机器语言中的 01 代码,这种用助记符号描述的指令系统称为汇编语言。为了把汇编语言编写的程序转换为机器语言程序,人们开发出了称为“汇编程序”的“翻译程序”。汇编语言指令与机器语言指令是一一对应的,因此,汇编语言也是与具体计算机硬件相关。汇编语言除了可读性比机器语言好外,同样也存在机器语言的缺点,尤其是描述问题的方式与人们习惯相差太远。

3. 高级语言

为了提高程序开发效率,针对机器语言和汇编语言的缺点,20 世纪 50 年代中期,IBM 公司的一个编程小组提出了一个设想,能否用数学公式来编写人们想要进行的计算,然后再让计算机将这些公式翻译(解释)为机器语言呢?最终该团队于 1955 年完成了 ForTran (formulation translation)的最初版本,这是第一个高级编程语言。从此以后,各种高级语言

相继涌现,其中能引起广泛关注和使用的主要有 BASIC、ALGOL、COBOL、Pascal、C 语言等。在众多的高级语言中,最为流行,也是最成功的当数 C 语言,它既用来写应用软件,也可用来写系统软件(如 UNIX 操作系统等)。

与低级语言(机器语言、汇编语言)相比,高级语言的表达方式更接近人类自然语言的表述习惯,具有较好的可读性,并且不依赖于计算机的具体型号,具有良好的可移植性。

1.2 程序设计

程序设计是一门用计算机解决问题的科学,在掌握程序设计错综复杂的内容之前,有必要对什么是程序设计这一概念有一个感性的认识。

1.2.1 程序设计概念

程序设计的目的就是用计算机解决问题。用计算机解决问题大体上要经过两个步骤,第一步是通过分析问题构造出一个解决问题的算法(即解决问题的方法和步骤),这个过程称为算法设计;第二步是用一种程序设计语言(如 C 语言)将该算法表达为程序,这个过程称为编码。程序设计的两个步骤是相辅相成的,作为一个程序设计新手,只能从简单的问题入手,而简单问题的解决方法也较简单,所以算法设计阶段不会有什么困难,而 C 语言对于初学者来说是全新的,因此初学程序设计时,编码常常会是两个步骤中较为困难的阶段。但可以确定的是,随着对 C 语句语法的了解,随着编程实践的不断增长,编码会越来越简单。相反地,随着遇到的问题愈来愈复杂,算法设计会变得更加困难和更具有挑战性。

1.2.2 算法

算法泛指解决某一个问题的方法和步骤。事实上,做任何事情都有一定的方法和步骤,例如洗手,你要首先打开水龙头,把手淋湿,然后用肥皂抹手,最后再伸手冲水把肥皂沫洗净,并关上水龙头。这些步骤都是按一定的顺序进行的,缺一不可,顺序错了也不行。因此,从事各种工作和活动,都必须事先设计好步骤,然后按部就班地进行,才能避免产生错乱。算法是一种解决问题的策略,是人们对问题进行分析 and 抽象的结果。无论是数学公式,还是计算机程序,都属于算法的具体表现。但算法不等于程序,其含义和范围更广。算法是程序设计的灵魂。不了解算法就谈不上程序设计,因此对于程序设计人员来说,必须会设计算法,并且能根据算法编写程序。需要说明的是,不要认为只有“计算”的问题才有算法。

1.2.3 程序设计的步骤

对于初学编程的人来说,往往简单地把程序设计理解为编写一个程序,认为能根据实际问题直接用计算机语言编出一个程序就行了,这样理解是不全面的。事实上,程序是程序设计的最终产品,需要经过中间每一步的细致加工才能得到。如果企图一开始就编写出程序,往往会适得其反,达不到预想的结果。对于大型的程序,由算法设计和编码两步骤组成只是大略说法,严格地说,程序设计一般应遵循以下步骤:

1. 分析问题

用计算机解决问题,首先要对问题进行分析,以便确定这个问题需要计算机做些什么?

如果没有把要解决的问题分析清楚之前就贸然开始编写程序,只能起到事倍功半的效果,而且很难得到满意的结果。因此,分析问题,弄清楚要解决的问题并给出问题的明确定义是解决问题的关键。

2. 系统设计

在弄清要解决的问题之后,就要考虑如何解决它,即如何做?从本质上看,用计算机解决问题的方式就是对数据进行处理。因此,首先要对问题进行抽象,抽取出能够反映本质特征的数据并对其进行描述,即给出问题的数据结构设计;然后考虑对数据如何进行操作以获得问题的结果,即进行算法设计。不同的程序设计方式在处理数据结构和算法两者之间的关系时是有所区别的。在面向过程程序设计中,把数据结构设计和算法设计分开考虑,而面向对象程序设计是把两者结合成对象来考虑。

3. 编码

在进行数据结构设计和算法设计时,往往采用某种与具体程序设计语言无关的语言(如伪代码或自然语言等)来描述算法。这样做的目的是为了从一开始就陷入程序设计语言的具体细节中。因为过多地涉及实现细节,不利于从较高抽象层次对问题本质进行考虑,并造成对设计过程难以把握和理解。当然,用伪代码(或自然语言或框图)描述的算法是不能被计算机执行的,必须用具体程序设计语言把它们表示出来,即编程实现。因此,用某种程序设计语言编写程序,也是对问题处理方案的描述,并且是最终的描述。程序文本保存在一个文件或多个文件中。包含程序文本的文件称为源文件,即源程序文件。

4. 测试与调试

源程序文件要经过编译程序翻译成等价的目标程序文件,并将一系列目标文件及库文件连接在一起生成可执行文件,程序才能被计算机执行。

程序编写好后,还需要进行调试和测试。只有经过调试的程序才能正式运行。所谓调试,是指找出程序中的错误并改正错误。因此,调试又称查错。而所谓测试,是指精心设计一批测试用例(包括输入数据和与之相应的预期输出结果),然后分别用这些测试用例运行程序,看程序的实际运行结果与预期输出结果是否一致,以尽可能多地发现程序中的错误。因此,测试的目的也是为了发现程序中的错误,而不是证明程序正确。调试和测试往往是交替进行的,通过测试发现程序中的错误,通过调试进一步找出错误的位置并改正错误。这个过程往往需要重复多次,这一步骤是程序设计中最困难的一步。

程序中的错误通常有三种:语法错误、逻辑错误和运行异常错误。语法错误是指源程序中存在违反C语法规则的地方,如语句后遗漏了分号“;”或忘记定义变量等,程序编译时可以发现这类错误,并会给出出错信息提示和出错位置,用户可以根据编译器提供的提示信息修改源程序。逻辑错误是指程序没有完成预期的功能,如源程序中错将“ $c = a \% b$ ”写成“ $c = a / b$ ”,或该用“ $==$ ”的地方却误用了“ $=$ ”等,编译程序时,编译器发现不了这类错误,但程序运行时,会产生错误的结果,甚至根本无法显示结果。遇到此类错误,用户往往需要通过设置断点,跟踪程序的运行过程等测试手段,才能发现它们。因此程序设计中最致命的错误是逻辑错误,因为它不太容易被发现,并且会导致程序不能正确地解决问题。在程序中找出并修正逻辑错误的过程称为程序调试,它是软件开发中不可缺少的一个环节。俗话说,“三分编程七分调试”,说明程序调试的工作量要比编程的工作量大得多。运行异常错误是指程序对程序运行环境的非正常情况考虑不足而导致的程序运行异常终止。逻辑错误和运

行异常错误可能是编程阶段导致的,也有可能是系统设计阶段或问题分析阶段的缺陷。这两类错误一般都要通过测试才能发现。

无论你采用何种测试手段,都只能发现程序有错,而不能证明程序正确。即使程序通过了良好的测试,总还是会有一些隐蔽的错误。作为一个编程者,任务就是尽可能地对程序进行彻底测试,不断找出并改正错误。

5. 整理文档资料

对于程序设计人员来说,平时的归纳和总结很重要。程序员应将平时的源程序和各种文字资料进行归类保存,以便今后的查询。最后还要编写使用和维护该程序的说明书,供程序用户参考。

6. 运行与维护

程序通过测试后就可交付使用了,但程序在使用中还需要不断得到维护。程序维护可分为正确性维护、完善性维护和适应性维护。程序即使经过大量测试,源代码中依然存在逻辑错误,这些错误会在程序的使用过程中不断暴露。当出现一些不常见的、未预料到的情况时,之前隐藏的异常错误就会导致程序运行失败。当一个程序使用了一段时间后,客户会期望程序能做些别的事情,对于用户提出的完善和扩充程序功能的要求,程序开发者应考虑响应这些要求。无论是排除逻辑错误还是要完善程序功能,在任何情况下都需要有人查看程序,做出必要的修改,并确保这些修改能使程序更好地工作。

如果对程序的功能进行了较大的更改,应发布程序的新版本。

1.3 C语言发展和C++简介

1.3.1 C语言发展简述

1972年,为了编写UNIX操作系统,贝尔实验室的D. M. Ritchie设计并实现了C语言。后来,C语言被多次进行改进,使其逐渐成熟,并先后移植到各种计算机上,现在C语言已成为世界上使用最广泛的程序设计语言之一。在软件产业蓬勃发展的今天,越来越多的程序都是由C语言编写的。

C语言在其发展过程中,涌现了众多不同的版本,其中有3个重要的标准。一是Brian W. Kernighan和Dennis M. Ritchie于1978年合著的名著*The C Programming Language*,被称为标准C。二是美国国家标准委员会(ANSI)于1983年制定的ANSI C,它于1989年通过。ANSI C制订了C语言及其运行时函数库的标准,比原来的标准C有了很大的发展。1990年国际标准化组织(ISO)对ANSI C做了少量修改,并通过了该标准,有的书也将该标准称为C89或C90。ISO标准定期审查更新,1999年发布了第三个重要标准,简称C99。该标准增加了很多新特性,如增加了关键字、复数运算、数据类型的扩展以及提高对非英语字符集的支持等。标准化促使C语言在商业、学术等编程领域得到更广泛的应用。

此后,尽管C++、Visual C++开发如火如荼,C语言也并没有停滞不前,更没有被替代,虽然版本不断更新、升级,但C语言的特征未变,C语言仍然是C语言。

1.3.2 C++简介

当C语言程序达到一定的规模后,维护和修改显得相当困难。为了满足管理程序复杂性的需要,贝尔实验室的Bjarne Stroustrup博士于1979年开始对C语言进行了改进和扩充,并引入了面向对象程序设计的内容,最初取名为“带类的C”,1983年改名为C++。在经历了3次重大修订后,于1994年制定了标准C++草案,后又经不断完善,成为目前的C++,它具有以下特点:

(1) C++是C语言的超集。C++由两部分组成:一是过程性语言部分,这部分与C语言无本质区别;二是类和对象部分,这是C语言所没有的,它是面向对象程序设计的主体。

(2) C++充分保持了与C语言的兼容性,绝大多数C语言程序可以不经修改直接在C++环境中运行。

(3) C++仍然支持面向过程的程序设计,不仅是一种理想的结构化程序设计语言,又几乎全部包含了面向对象程序设计的特征。

(4) C++继承了C语言的高效率、灵活性好等优点。用Bjarne Stroustrup博士的话来说,C++使程序“结构清晰、易于扩展、易于维护而不失效率”。

(5) C++是一种标准化的、与硬件基本无关的、广泛使用的程序设计语言,具有很好的通用性和可移植性。C++程序通常无需修改,或稍作修改,即可在其他计算机系统上运行。

1.3.3 集成开发环境

本书将介绍两种集成开发环境:Visual C++ 6.0和Dev-C++ 5.11。它们既能开发C++语言程序,也能开发C程序,而且操作简单易学。本书所有例子的编程环境均在Visual C++ 6.0和Dev-C++ 5.11中通过测试。编程者只需选择其中一种作为自己的编程工具,然后学习它的使用方法(见1.5节或1.6节)。

1.4 C语言程序的构成

C语言属于人造语言,因此C语言与自然语言之间有很多相似之处。自然语言(如英文)由句子、单词和字母组成,C语言也由基本符号(字符)构成一系列单词(语法元素),由多个单词构成句子(语句),再由多个语句构成程序。不同的是C语言具有较严格的语法规则,在语义上也不像自然语言那样具有多义性,程序文本所代表的语义是单一的、确定的。正像写文章时分章节、段落和层次一样,C程序也具有层次结构,但无论程序规模大小,C程序都是由函数组成的,而函数又是由语句组成的。

本节将从编写简单的输出程序入门,循序渐进地学习,使读者从整体上对C语言程序的基本结构有一个感性认识,下面首先学习简单程序的模板。

C语言的程序由一个或多个函数组成,main函数是必需的。初学者应从学习编写只有一个函数的程序开始,也就是只有main函数的程序。下面是main函数的模板,编程语句写在return 0前的空白处。

```
int main()  
{
```

```

    语句
    return 0;
}

```

main 是函数的名字,也称为主函数。无论程序中有多少个函数,执行程序时,系统自动从 main 函数开始。花括号{ }界定函数内容的起始和结束,把编写的程序放在{}内。

花括号中的 return 0 可以结束程序的运行,并且将 0 返回给操作系统,0 告诉操作系统程序是正常结束的。因此可见,main 函数是程序的起始和结束,它是每个程序的必写部分。

下面学习一个简单的输出程序,它是一个经典的入门程序,是第一本 C 语言的书 *The C Programming Language* 的第一个入门例题。

【例 1.4.1】 输出一行字符: hello, world。

```

#include <stdio.h>
int main()
{
    printf("hello, world");
    return 0;
}

```

说明:

(1) 在 main 函数中仅添加了一条语句,其中的 printf 是 C 语言的标准输出函数,它将输出(" ")中的内容。结果显示 hello, world。

(2) C 语言没有输入/输出命令,输入/输出的功能由 C 标准输入/输出函数库中的函数实现,printf 就是其中常用的重要函数。在使用标准输入/输出库函数之前,必须添加预处理命令: #include <stdio.h>,其中的 stdio.h 文件内有库函数的信息,include 命令将函数信息包含进本程序中(见 7.3 节)。

(3) 每个语句的最后以分号结束。注意:程序设计中应使用英文的标点符号,不能使用中文的标点符号。

【例 1.4.2】 输出计算结果。

```

#include <stdio.h>
int main()
{
    printf("%d", 2 * 5);
    return 0;
}

```

科学计算是程序设计的必备功能,输出计算结果也依赖于 printf 函数。printf 中 f 是 format 的意思,指按格式打印输出()中的内容。printf 函数的完整功能见 3.3.1 节,本例学习输出整数。

说明:

(1) printf 函数的一般格式如下:

```
printf("格式控制字符串",表达式列表);
```

功能:按格式控制字符串中的格式依次输出表达式的值。%d 称为整型格式控制符,输

出整数表达式的值。

本例输出结果为 10, 如果需要输出 n 个整数表达式的值, 就需要 n 个 `%d`。例如,

```
printf("%d, %d, %d", 5, 2 * 5, 5 * 2 * 5);
```

输出结果如下:

```
5, 10, 50
```

格式控制字符串中非格式符原样输出, 如 `%d` 之间的逗号。再如,

```
printf("length is : %d, width is : %d, area is : %d", 5, 2 * 5, 5 * 2 * 5);
```

输出结果如下:

```
length is : 5, width is : 10, area is : 20
```

(2) 为了使内容在下一行输出, 可以在格式控制字符串中添加 `\n` (换行符, 见表 2.3.1)。`\n` 使光标换行, 其后续的输出从新的一行开始。例如:

```
printf("The result is: \n%d", 2 * 5);
```

输出结果如下:

```
The result is:
10
```

下面的例题是由 2 个函数组成的程序, 包含 C 程序更多的构成元素。

【例 1.4.3】 已知圆柱体的半径 r 、高 h , 求其体积 v 。

```
/*
file: compute volume
该程序是用于计算圆柱体体积
*/
#include <stdio.h>
#define PI 3.14159
float volume(float r, float h);
int main()
{
    float radius, height, vol;
    printf("input radius, height: ");
    scanf("%f %f", &radius, &height);
    vol = volume(radius, height);
    printf("vol = %f", vol);
    return 0;
}
/*
* 函数: volume
* 用法: v = volume(r, h);
* 该函数的功能是计算半径为 r、高为 h 的圆柱体的体积
*/
float volume(float r, float h)
```

} 程序注释

} 包含库文件

} 常量定义

} volume 函数原型声明

} 主程序

} 函数注释