



21世纪高等院校规划教材
程序设计类

C语言程序设计

主编 林军



中国水利水电出版社
www.waterpub.com.cn

21 世纪高等院校规划教材

C 语言程序设计

主 编 林 军



中国水利水电出版社

www.waterpub.com.cn

www.waterpub.com.cn

内 容 提 要

本书是为了满足广大学生学习 C 语言程序设计的需要,结合高等学校 C 语言程序设计课程的教学特点,由高等学校长期从事计算机语言课程教学的第一线教师编写而成,内容涵盖了《全国计算机等级考试大纲(2009 年版)》中对 C 语言程序设计要求的知识点。

全书共 11 章,内容包括 C 语言基础知识、数据类型与运算规则、顺序结构的程序设计、选择结构的程序设计、循环结构的程序设计、数组、字符串处理、函数、指针、结构体与共用体和文件。各章均有精心设计的大量习题。本书理论与实践紧密结合,书中的例子具有实用性,内容安排合理,重点突出。

本书由浅入深、循序渐进、前后呼应、通俗易懂、图文并茂、内容丰富,书中的所有程序都在 Turbo C 2.0 环境下调试成功。与本书配套的《C 语言程序设计实验指导与习题解答》中包括 C 语言程序设计实验上机操作指导、实验、主教材的习题答案以及课程设计指导等。

本书既可以作为高等学校本科及专科学生 C 语言程序设计的教材,又可以作为全国计算机等级考试二级——C 语言程序设计的培训教材,同时也可以作为自学者的参考书。

本书配有电子教案,读者可以到中国水利水电出版社或万水书苑网站免费下载,网址:
<http://www.waterpub.com.cn/softdown/>或 <http://www.wsbookshow.com>。

图书在版编目(CIP)数据

C语言程序设计 / 林军主编. -- 北京:中国水利水电出版社, 2010.2
21世纪高等院校规划教材
ISBN 978-7-5084-7080-1

I. ①C… II. ①林… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2009)第231106号

策划编辑:雷顺加 责任编辑:李 炎 加工编辑:李 刚 封面设计:李 佳

书 名	21 世纪高等院校规划教材 C 语言程序设计
作 者	主 编 林 军
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	184mm×260mm 16 开本 18.5 印张 476 千字
版 次	2010 年 2 月第 1 版 2010 年 2 月第 1 次印刷
印 数	0001—4000 册
定 价	30.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换
版权所有·侵权必究

序

随着计算机科学与技术的飞速发展,计算机的应用已经渗透到国民经济与人们生活的各个角落,正在日益改变着传统的人类工作方式和生活方式。在我国高等教育逐步实现大众化后,越来越多的高等院校会面向国民经济发展的第一线,为行业、企业培养各级各类高级应用型专门人才。为了大力推广计算机应用技术,更好地适应当前我国高等教育的跨越式发展,满足我国高等院校从精英教育向大众化教育的转变,符合社会对高等院校应用型人才需求的各类要求,我们成立了“21世纪高等院校规划教材编委会”,在明确了高等院校应用型人才培养模式、培养目标、教学内容和课程体系的框架下,组织编写了本套“21世纪高等院校规划教材”。

众所周知,教材建设作为保证和提高教学质量的重要支柱及基础,作为体现教学内容和教学方法的知识载体,在当前培养应用型人才中的作用是显而易见的。探索和建设适应新世纪我国高等院校应用型人才需求体系需要的配套教材已经成为当前我国高等院校教学改革和教材建设工作面临的紧迫任务。因此,编委会经过大量的前期调研和策划,在广泛了解各高等院校的教学现状、市场需求,探讨课程设置、研究课程体系的基础上,组织一批具备较高的学术水平、丰富的教学经验、较强的工程实践能力的学术带头人、科研人员和主要从事该课程教学的骨干教师编写出一批有特色、适用性强的计算机类公共基础课、技术基础课、专业及应用技术课的教材以及相应的教学辅导书,以满足目前高等院校应用型人才培养的需要。本套教材消化和吸收了多年来已有的应用型人才培养的探索与实践成果,紧密结合经济全球化时代高等院校应用型人才培养工作的实际需要,努力实践,大胆创新。教材编写采用整体规划、分步实施、滚动立项的方式,分期分批地启动编写计划,编写大纲的确定以及教材风格的定位均经过编委会多次认真讨论,以确保该套教材的高质量和实用性。

教材编委会分析研究了应用型人才与研究型人才在培养目标、课程体系和内容编排上的区别,分别提出了3个层面上的要求:在专业基础类课程层面上,既要保持学科体系的完整性,使学生打下较为扎实的专业基础,为后续课程的学习做好铺垫,更要突出应用特色,理论联系实际,并与工程实践相结合,适当压缩过多过深的公式推导与原理性分析,兼顾考研学生的需要,以原理和公式结论的应用为突破口,注重它们的应用环境和方法;在程序设计类课程层面上,把握程序设计方法和思路,注重程序设计实践训练,引入典型的程序设计案例,将程序设计类课程的学习融入案例的研究和解决过程中,以学生实际编程解决问题的能力为突破口,注重程序设计算法的实现;在专业技术应用层面上,积极引入工程案例,以培养学生解决工程实际问题的能力为突破口,加大实践教学内容的比重,增加新技术、新知识、新工艺的内容。

本套规划教材的编写原则是:

在编写中重视基础,循序渐进,内容精炼,重点突出,融入学科方法论内容和科学理念,反映计算机技术发展要求,倡导理论联系实际和科学的思想方法,体现一级学科知识组织的层次结构。主要表现在:以计算机学科的科学体系为依托,明确目标定位,分类组织实施,兼容互补;理论与实践并重,强调理论与实践相结合,突出学科发展特点,体现学科发展的内在规律;教材内容循序渐进,保证学术深度,减少知识重复,前后相互呼应,内容编排合理,整体

结构完整;采取自顶向下设计方法,内涵发展优先,突出学科方法论,强调知识体系可扩展的原则。

本套规划教材的主要特点是:

(1) 面向应用型高等院校,在保证学科体系完整的基础上不过度强调理论的深度和难度,注重应用型人才的专业技能和工程实用技术的培养。在课程体系方面打破传统的研究型人才培养体系,根据社会经济发展对行业、企业的工程技术需要,建立新的课程体系,并在教材中反映出来。

(2) 教材的理论知识包括了高等院校学生必须具备的科学、工程、技术等方面的要求,知识点不要求大而全,但一定要讲透,使学生真正掌握。同时注重理论知识与实践相结合,使学生通过实践深化对理论的理解,学会并掌握理论方法的实际运用。

(3) 在教材中加大能力训练部分的比重,使学生比较熟练地应用计算机知识和技术解决实际问题,既注重培养学生分析问题的能力,也注重培养学生思考问题、解决问题的能力。

(4) 教材采用“任务驱动”的编写方式,以实际问题引出相关原理和概念,在讲述实例的过程中将本章的知识点融入,通过分析归纳,介绍解决工程实际问题的思想和方法,然后进行概括总结,使教材内容层次清晰,脉络分明,可读性、可操作性强。同时,引入案例教学和启发式教学方法,便于激发学习兴趣。

(5) 教材在内容编排上,力求由浅入深,循序渐进,举一反三,突出重点,通俗易懂。采用模块化结构,兼顾不同层次的需求,在具体授课时可根据各校的教学计划在内容上适当加以取舍。此外还注重了配套教材的编写,如课程学习辅导、实验指导、综合实训、课程设计指导等,注重多媒体的教学方式以及配套课件的制作。

(6) 大部分教材配有电子教案,以使教材向多元化、多媒体化发展,满足广大教师进行多媒体教学的需要。电子教案用 PowerPoint 制作,教师可根据授课情况任意修改。相关教案的具体情况请到中国水利水电出版社网站 www.waterpub.com.cn 下载。此外还提供相关教材中所有程序的源代码,方便教师直接切换到系统环境中教学,提高教学效果。

总之,本套规划教材凝聚了众多长期在教学、科研一线工作的教师及科研人员的教学科研经验和智慧,内容新颖,结构完整,概念清晰,深入浅出,通俗易懂,可读性、可操作性和实用性强。本套规划教材适用于应用型高等院校各专业,也可作为本科院校举办的应用技术专业的课程教材,此外还可作为职业技术学院和民办高校、成人教育的教材以及从事工程应用的技术人员的自学参考资料。

我们感谢该套规划教材的各位作者为教材的出版所做出的贡献,也感谢中国水利水电出版社为选题、立项、编审所做出的努力。我们相信,随着我国高等教育的不断发展和高校教学改革的不深入,具有示范性并适应应用型人才培养的精品课程教材必将进一步促进我国高等院校教学质量的提高。

我们期待广大读者对本套规划教材提出宝贵意见,以便进一步修订,使该套规划教材不断完善。

21 世纪高等院校规划教材编委会

2004 年 8 月

前 言

本书是根据高等学校 C 语言程序设计课程教学特点,由高等学校长期从事 C 语言程序设计课程教学与科研开发的一线教师编写而成,书中内容涵盖了《全国计算机等级考试大纲(2009 年版)》中对 C 语言程序设计要求的知识点。

全书共 11 章,内容包括 C 语言基础知识、数据类型与运算规则、顺序结构的程序设计、选择结构的程序设计、循环结构的程序设计、数组、字符串处理、函数、指针、结构体与共用体和文件。在每一章都配有精心安排的练习题,这些习题包括了读者应掌握的各章基本知识以及应掌握的程序设计基本方法,并且在本书的配套教材中有所有习题的参考答案,有利于读者练习、巩固与提高。书中的所有程序都在 Turbo C 2.0 环境下调试成功。

本书的编写考虑到了从未接触过计算机的学生学习程序设计的需求,在介绍程序设计方法时,使用学生所熟悉和实用的例子、直观的示意图、程序流程图以及简单明了的书面语言,突出程序设计的基本思想,强调算法的重要性,引导学生逐步掌握知识,将教学法融入书中。由于 C 语言程序设计的内容十分丰富,要在有限的学时内讲授全部内容是困难的,因此,本书本着实用、够用、重基础的原则,对介绍的内容做了必要的取舍,突出重点,从而保证了教学内容的完整性,使学生学习了基础知识后,具备自学书中未介绍的 C 语言程序设计内容的能力,使学生通过本课程的学习,能够融会贯通,很快学会和掌握其他高级语言程序设计的方法,并且为数据结构、汇编语言、面向对象的程序设计以及其他后续课程的学习,打下良好的基础。实践性强是 C 语言程序设计课程的特点之一,因此,必须通过上机实验,才能很好地掌握本课程的知识。与本书配套的《C 语言程序设计实验指导与习题解答》中包括 C 语言程序设计实验上机操作指导、实验、主教材的习题答案以及课程设计指导等。

本书适合用做大专院校的计算机专业或非计算机专业的本科与专科学生 C 语言程序设计课程的教材,教师可以根据教学大纲的具体要求来决定哪些章节需要讲解。本书可以帮助初学者快速入门,是自学者的好帮手,有利于读者参加全国计算机等级考试,可以作为二级 C 语言程序设计考试的培训教材,也可以作为科技人员的参考用书。

本书是在日常教学的基础上,对 C 语言程序设计讲义进行修改整理后编写而成的。本书由林军任主编,叶雪军、李祥、孟祥宏任副主编,参加本书编写工作的还有:陈昌彦、陈婕、邓沌华、谭生龙、邓伟萍、林民、宋莺、姜勇杰、姚竞、王文剑、余铮、钟臣、刘畅、雷正旺、杨佩佩、陈希、周萍和文静等。

由于编者水平有限,书中难免有疏漏不当之处,恳请广大读者批评指正。

编 者

2009 年 12 月

目 录

序	1
前言	1
第1章 C语言基础知识	1
1.1 C语言概述	1
1.1.1 C语言简介	1
1.1.2 C语言的特点	2
1.2 程序设计基础知识	2
1.2.1 程序设计简介	2
1.2.2 算法简介	3
1.2.3 程序基本结构及其流程图表示	5
1.3 C语言程序的基本构成	8
1.3.1 一个简单的C语言程序	8
1.3.2 C语言程序的构成	9
1.3.3 关键字	10
1.3.4 标识符	10
1.3.5 语句	10
1.3.6 标准库函数	11
1.4 C语言程序的运行	11
习题一	13
第2章 数据类型与运算规则	16
2.1 概述	16
2.2 常量与变量	17
2.2.1 常量	17
2.2.2 变量	20
2.3 算术运算与赋值运算	24
2.3.1 C语言中的运算规则	24
2.3.2 算术运算符与算术表达式	25
2.3.3 自增、自减运算	26
2.3.4 赋值运算符与赋值表达式	26
2.3.5 组合赋值运算符与组合赋值表达式	27
2.4 关系运算与逻辑运算	28
2.4.1 关系运算符与关系表达式	28
2.4.2 逻辑运算符与逻辑表达式	29
2.4.3 条件运算符与条件表达式	29
2.5 位运算	30
2.5.1 位运算符	30
2.5.2 按位逻辑运算	30
2.5.3 移位运算	31
2.5.4 位运算组合赋值运算符	32
2.6 其他运算	32
2.6.1 逗号运算符	32
2.6.2 “.”和“→”运算符	32
2.6.3 “()”和“[]”运算符	33
2.6.4 “*”和“&”运算符	33
2.6.5 (类型)运算符	33
2.6.6 sizeof运算符	33
2.7 混合运算及数据类型转换	34
习题二	36
第3章 顺序结构的程序设计	39
3.1 C语句概述	39
3.1.1 流程控制语句	39
3.1.2 表达式语句	40
3.1.3 函数调用语句	40
3.1.4 复合语句	40
3.1.5 空语句	40
3.2 赋值语句	40
3.3 数据输出	41
3.3.1 字符输出函数 putchar()	41
3.3.2 格式化输出函数 printf()	42
3.3.3 格式字符串	43
3.4 数据输入	45
3.4.1 字符输入函数 getchar()	45
3.4.2 格式化输入函数 scanf()	45
3.4.3 格式字符串	45
3.5 应用举例	48
习题三	51
第4章 选择结构的程序设计	55
4.1 概述	55

4.2 if 语句	55	7.2.4 字符串的输入与输出函数	145
4.2.1 简单的 if 语句	55	7.3 字符串处理函数	148
4.2.2 双分支 if 语句——if-else 语句	57	7.3.1 求字符串长度	148
4.2.3 多分支 if 语句——if-else-if 语句	60	7.3.2 字符串复制	149
4.3 switch 语句	66	7.3.3 字符串连接	150
4.4 goto 语句	69	7.3.4 字符串比较	151
4.5 应用举例	70	7.3.5 大写字母转换为小写字母	152
习题四	75	7.3.6 小写字母转换为大写字母	152
第 5 章 循环结构的程序设计	84	7.4 二维字符数组	153
5.1 概述	84	7.4.1 二维字符数组的定义	153
5.2 while 语句	84	7.4.2 二维字符数组的初始化	153
5.3 do-while 语句	88	7.4.3 二维字符数组的引用	154
5.4 for 语句	90	习题七	155
5.5 break 语句与 continue 语句	95	第 8 章 函数	159
5.5.1 break 语句	95	8.1 概述	159
5.5.2 continue 语句	96	8.2 函数的语法	161
5.6 3 种循环语句的比较	97	8.2.1 函数定义的一般格式	161
5.7 循环的嵌套	98	8.2.2 自定义函数的声明	166
5.8 应用举例	101	8.2.3 函数参数和函数的值	167
习题五	107	8.2.4 函数的调用	169
第 6 章 数组	114	8.3 数组作为函数参数	170
6.1 数组的基本概念	114	8.4 函数的嵌套调用	173
6.2 一维数组	115	8.5 函数的递归调用	175
6.2.1 一维数组的定义	115	8.5.1 递归的概念	175
6.2.2 一维数组的引用	115	8.5.2 递归程序设计应用举例	175
6.2.3 一维数组的初始化	116	8.6 局部变量和全局变量	178
6.2.4 一维数组的应用	116	8.6.1 局部变量	178
6.3 二维数组	129	8.6.2 全局变量	178
6.3.1 二维数组的定义	129	8.7 变量的存储类别	179
6.3.2 二维数组的引用	130	8.7.1 动态存储方式与静态存储方式	179
6.3.3 二维数组的初始化	130	8.7.2 auto 变量	180
6.3.4 二维数组的应用	131	8.7.3 用 static 声明的局部变量	180
习题六	137	8.7.4 register 变量	181
第 7 章 字符串处理	142	8.7.5 用 extern 声明的外部变量	182
7.1 概述	142	8.8 工程文件	184
7.2 一维字符数组	142	8.9 编译预处理	184
7.2.1 一维字符数组的定义	142	8.9.1 宏定义	185
7.2.2 一维字符数组的初始化	143	8.9.2 文件包含	187
7.2.3 一维字符数组的引用	144	8.9.3 条件编译	188

习题八	190	10.6.2 链表的基本运算	240
第9章 指针	196	10.7 共用体	244
9.1 指针的基本概念	196	10.7.1 共用体	245
9.1.1 指针变量的定义	196	10.7.2 共用体与结构体的区别	245
9.1.2 指针变量的初始化	197	10.7.3 共用体应用举例	247
9.1.3 使用指针引用变量	198	10.8 类型定义——typedef	248
9.2 指针与函数	198	习题十	248
9.2.1 指针作函数的参数	198	第11章 文件	252
9.2.2 函数返回指针	202	11.1 C语言文件概述	252
9.3 指针与数组	203	11.1.1 文件的分类	252
9.3.1 通过指针引用一维数组	203	11.1.2 文件的处理	253
9.3.2 指针基本运算	206	11.1.3 文件指针	253
9.3.3 通过指针引用二维数组元素	210	11.2 文件的打开与关闭	254
9.4 指针与字符串	216	11.3 文件的检测	256
9.4.1 字符数组与字符指针	216	11.3.1 文件结束检测函数 feof()	256
9.4.2 应用举例	217	11.3.2 文件读写出错检测函数 ferror()	256
9.5 指针数组	218	11.3.3 exit()函数	257
9.6 指向指针的指针	220	11.4 文件的顺序读写	257
习题九	222	11.4.1 单字符读写函数 fgetc()和 fputc()	257
第10章 结构体与共用体	226	11.4.2 字符串的读写函数 fgets()和 fputs()	260
10.1 概述	226	11.4.3 数据块读写函数 fread()和 fwrite()	262
10.2 结构类型声明	227	11.4.4 格式化读写函数 fscanf()和 fprintf()	264
10.3 结构变量	228	11.5 文件的随机读写	265
10.3.1 结构变量的定义	228	11.5.1 rewind()函数	265
10.3.2 结构变量的初始化	230	11.5.2 fseek()函数	266
10.3.3 结构变量的引用	231	11.5.3 ftell()函数	266
10.4 结构数组	233	习题十一	267
10.4.1 结构数组的定义	233	附录A 运算符	270
10.4.2 结构数组的初始化	234	附录B C语言常用库函数	272
10.4.3 结构数组应用举例	235	附录C ASCII码表	279
10.5 结构指针	236	附录D 关于C语言程序的书写格式	281
10.5.1 指向结构变量的指针	236	参考文献	286
10.5.2 指向结构数组的指针	237		
10.5.3 结构与函数	238		
10.6 链表处理——结构指针的应用	239		
10.6.1 链表概述	239		

第1章 C语言基础知识

本章导读

本章首先简单介绍C语言及其特点,并介绍语言、程序、机器语言、汇编语言、高级语言等基本概念以及程序设计的步骤。然后,简单介绍算法和算法的表示方法,并通过几个具体例子来说明怎样使用流程图和N-S流程图表示算法。本章还介绍了C语言、C语言程序及语句的构成。本章通过一个简单的C语言程序的例子使学生初步认识和了解C语言程序。最后,介绍C语言程序的上机执行过程,阐述程序的编辑、编译、链接和运行等基本知识。

本章要点

- ◆ 1.1 C语言概述
- ◆ 1.2 程序设计基础知识
- ◆ 1.3 C语言及程序的基本构成
- ◆ 1.4 C语言程序的运行

1.1 C语言概述

1.1.1 C语言简介

C语言是一种很有特色的计算机高级语言,目前在国内外十分流行,并被广泛应用,显示了很好的发展前景和极强的生命力。

C语言是在一种B语言的基础上发展起来的,于20世纪70年代初,由美国贝尔实验室研究人员共同开发完成。

C语言问世以后,经历了一段不断改进和发展的历程。最初C语言是为描述和实现UNIX操作系统而提供的一种工具语言,随着UNIX操作系统的广泛应用,C语言也得到迅速推广。1978年,Brian W. Kernighan和Dennis M. Ritchie出版了著名的《The C Programming Language》一书,书中介绍的C语言是当时UNIX操作系统上的版本,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。后来,C语言又得到多次改进,出现了多种版本。为了改变这一状况,建立统一的标准,1983年,美国国家标准化协会(American National Standard Institute, ANSI)根据各种C语言版本对C语言进行了改进和扩充,制定了ANSI C标准。1987年,美国国家标准化协会又颁布了新标准,称为87ANSI C标准。1990年,C语言成为国际标准化组织(International Standards Organization, ISO)

通过的标准语言。

1.1.2 C 语言的特点

C 语言因其鲜明的特色，得以存在和发展。下面从不同角度总结 C 语言的特点和优点。

(1) C 语言具有综合性。C 语言是一种综合性语言，C 语言将高级语言的功能同汇编语言的功能结合起来，使 C 语言既有高级语言的功能，又具有汇编语言那样低级语言的许多功能。C 语言的这种同时具有高级语言和汇编语言功能的双重性，使它既可作为系统设计语言编写操作系统，也可作为应用程序设计语言编写应用程序。

(2) C 语言具有简洁性。C 语言十分简洁，使用方便，易于掌握。C 语言一共有 32 个关键字，9 种控制语句，程序书写格式自由。

(3) C 语言具有运算符的丰富性。C 语言一共有 34 种运算符，运算类型十分丰富，表达式类型多样化，可以实现在其他高级语言中难以实现的运算。众多的运算符和语句类型使 C 语言具有超常的表达能力。

(4) C 语言具有较好的可移植性。C 语言程序基本上可以不做任何修改，就能在各种不同型号的计算机和各种操作系统上运行。

(5) C 语言具有执行程序的高效性。C 语言源程序转换生成的目标代码质量高，一般只比汇编语言程序生成的目标代码效率低 10%~20%。

(6) C 语言是一种模块化的程序设计语言。C 语言的主要结构成分是函数，程序的函数结构十分便于将一个大的程序按功能分割成一些相对独立的功能模块，使每个模块都成为功能单一、结构清晰、容易理解的函数。这种函数结构的特点使 C 语言便于实现模块化的程序结构。

(7) C 语言是一种结构化的程序设计语言。C 语言提供了三种基本结构的语句，从而可以很方便地实现自顶向下、逐步求精的结构化程序设计技术。而且，C 语言提供了定义和调用函数的功能，并能将一个程序分解为多个程序文件，从而很容易实现程序的模块化设计。除此之外，C 语言还提供了丰富的数据类型与数据结构，具有很强的数据处理能力，这些都为结构化程序设计提供了强有力的工具。

1.2 程序设计基础知识

1.2.1 程序设计简介

1. 计算机语言

人和人交流用的是双方都能听懂的自然语言，同样，人和计算机交流也要用人和计算机都容易接受和理解的语言，这就是计算机语言。计算机语言是根据计算机的特点而编制的，它没有自然语言那么丰富多样，而只是有限规则的集合，所以它简单易学。

计算机语言可以分为 4 代，分别是机器语言、汇编语言、高级语言和面向对象语言。其中，机器语言和汇编语言属于低级语言。低级语言是面向机器的语言。机器语言是表示成二进制码形式的机器基本指令集。其特点是计算机可以直接识别，但由于使用机器语言设计的程序是由一串“0”或“1”的二进制代码——机器码组成的，因此，程序设计人员难以理解，工作效率低。汇编语言是用助记符、符号地址和操作数等表示的指令集。其特点是与特定的机器有关、程序执行的效率高，比起机器语言有容易理解、便于阅读、方便找出程序中的错误以及不需要熟记机器码等优点，但使用起来还是比较复杂，要求程序设计人员必须对硬件有很好的了

解。计算机不能识别汇编语言程序,必须将汇编语言源程序翻译成目标程序后才能在计算机上运行。高级语言是使用日常的英文单词、符号和数字,以一定的语法形式表示的语句集合。其特点是在一定程度上与具体机器无关,简单易学,使用方便,程序的可读性、可移植性和可维护性强。高级语言程序也需要翻译成计算机能够识别的机器码,才能在计算机上运行。一般来说经翻译后所产生的可执行程序往往效率较汇编语言低。高级语言是面向过程的语言。高级语言种类很多,例如,ALGOL 60、FORTRAN、BASIC、Pascal、C和COBOL等都属于高级语言。面向对象语言是以对象作为基本程序结构单位的程序设计语言,所有操作都以对象为基础,它是由面向过程语言发展而来的。面向对象的语言也有很多种,例如,C++、Java、Visual FoxPro和Visual Basic等都属于面向对象的程序设计语言。目前,面向对象的语言应用十分广泛,掌握好高级语言程序设计,可以为学习面向对象语言的程序设计打下基础。

2. 程序

程序是计算机指令的序列,编制程序的工作就是为计算机安排指令序列。通常一个计算机程序是用某种程序语言书写的。一般来说,高级语言程序是有限的语句集合。在C语言中,程序是由一个主函数main()和若干个自定义函数构成的,程序也可以只有一个主函数。主函数和这些自定义函数又是由若干个语句组成的。总之,使用语句设计的程序或是使用语句设计的函数所构成的程序,可以在计算机上运行,从而指挥计算机完成特定的操作,以满足各种需要。可以设计程序完成各种科学计算的工作,也可以设计程序对非数值的数据进行各种处理。

3. 程序设计的步骤

一个简单的程序设计一般包括以下4个步骤:

- (1) 分析问题,建立数学模型。
- (2) 确定数据结构和算法。
- (3) 编制程序。
- (4) 调试程序。

在设计程序时,首先必须对需要解决的问题进行分析,建立研究对象的数学模型。数学模型是对实际问题的数学描述。当需要从定量的角度分析和解决一个实际问题时,就要通过研究与分析,用数学符号和语言将其表述为数学式子,也就是数学模型。通过建立数学模型,使得实际问题可以在计算机上得以解决。建立数学模型是首要的也是关键的一个步骤。然后就是确定数据结构和算法,数据结构和算法是程序设计的两大要素,确定了算法,程序设计就有了可行性,如果没有算法,程序设计就无从下手。完成了以上两个步骤之后,就可以设计程序了,也就是编制程序。编制好的程序还不能立即投入运行,必须经过调试,才能运行。调试程序就是发现和纠正程序中错误的过程,具体地说,就是通过一定的方法,找出程序中存在的语法错误和逻辑错误,然后对程序进行修改,排除这些错误,直至程序能够在计算机上正常运行,得出正确的运行结果。一般来说,一个程序要经过反复的调试,才能够正常运行。

1.2.2 算法简介

1. 算法的概念

算法和数据结构是程序设计的两大要素。算法是指解决问题的一种方法或者一个过程,是计算机操作步骤的集合。数据结构是对数据的描述和组织形式,是指相互间有关联的数据元素的集合。著名计算机科学家Niklaus Wirth在1975年出版了著作《Algorithms + Data Structures = Programs》,由此可以看出算法在程序中的重要性。算法和数据结构在教学计划中,是两门专业课程,内容很多,在这里只是对算法做一个简单的介绍。

2. 算法的基本特征

算法是指为解决某个特定问题而采取的确定方法和有限步骤。一个算法应具有以下 5 个基本特征：

- (1) 有穷性。有穷性是指一个算法必须在执行有限个操作步骤后终止。
- (2) 确定性。确定性是指算法中每一步的含义必须是确切的，不可出现二义性。
- (3) 有效性。有效性是指算法中的每一步操作都能有效执行，一个不可执行的操作是无效的。例如一个数被 0 除的操作就是无效的，应当避免这种操作。
- (4) 有零个或多个输入。所谓输入是指在算法开始之前所需要的初始数据。这些输入数据的多少取决于特定的问题。有些算法也可以没有输入。

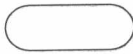
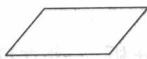
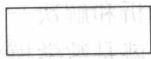
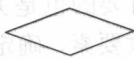
(5) 有一个或多个输出。所谓输出是指与输入有某种特定关系的量，在一个完整的算法中至少会有一个输出。

3. 算法的表示

有许多表示算法的方法。常用的描述方法有自然语言、程序语言、流程图、N-S 图、PAD 图和伪代码等。下面简单介绍流程图和 N-S 结构流程图。

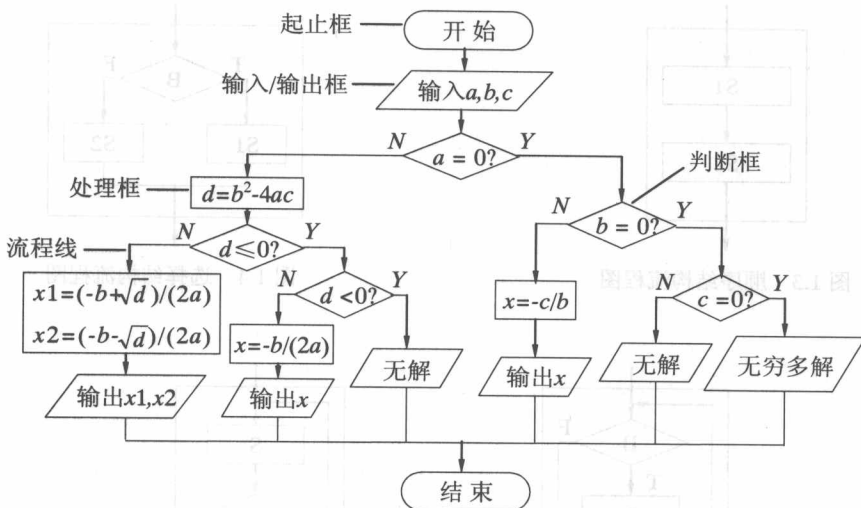
(1) 流程图。流程图是表示算法的一种常用工具。流程图又称为框图，用它来表示算法，形象直观，简单易懂。美国国家标准化协会 ANSI 规定了一些常用的流程图符号，已为世界各国程序工作者普遍采用。流程图的符号、符号名称和功能如表 1.1 所示。

表 1.1 标准流程图符号

名称	符号	功能
起止框		表示算法的开始和结束
输入/输出框		表示输入或者输出操作
处理框		表示算法中的各种处理操作
判断框		表示算法中的条件判断操作
注释框		对算法中某个操作进行说明
流程线		表示算法的执行方向
连接点		表示流程图的衔接，使用 2 个文字相同的连接点来表示位于不同位置上的同一点

下面使用一个解一元二次方程 $ax^2+bx+c=0$ 的具体例子来说明这些流程图符号的含义。图 1.1 是解一元二次方程 $ax^2+bx+c=0$ 的流程图。

(2) N-S 结构流程图。1973 年，美国学者 Ike Nassi 和 Ben Shneiderman 提出了一种新的流程图形式，在这种流程图中完全去掉了流程线，全部算法写在矩形框内，矩形框可以嵌套。这种流程图被称作 N-S 流程图。

图 1.1 解一元二次方程 $ax^2+bx+c=0$ 的流程图

(3) 用流程图表示算法举例。下面通过一个具体的例子来说明如何用流程图来表示算法。

例 1.1 已知圆的半径，求圆的面积。其算法的流程图与 N-S 流程图如图 1.2 所示。

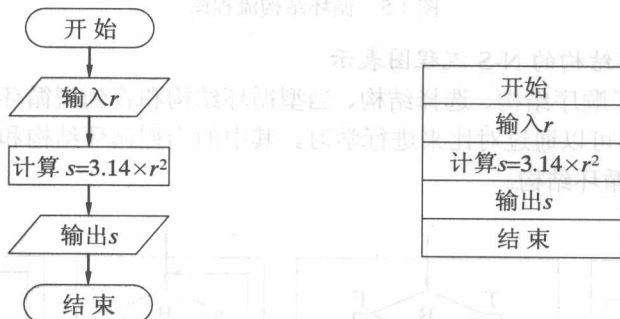


图 1.2 求圆的面积的流程图与 N-S 流程图

1.2.3 程序基本结构及其流程图表示

结构化程序设计的基本特征之一是使用顺序结构、选择结构和循环结构这三种基本结构来构成程序。下面简单介绍程序的三种基本结构以及怎样使用流程图来表示程序的三种基本结构。

1. 程序的三种基本结构

(1) 顺序结构。顺序结构的程序是顺序地执行一个个语句，不改变程序的执行方向。图 1.3 是顺序结构流程图，在图中，使用 S1 和 S2 表示语句。

(2) 选择结构。选择结构的程序是根据条件判断的结果，来决定程序的执行方向，通过选择结构的程序，可以构成分支。图 1.4 是一个双分支选择结构流程图，图中使用 B 表示判断条件，使用“T”表示条件成立 (true)，使用“F”表示条件不成立 (false)，也可以使用“Y”表示条件成立 (yes)，使用“N”表示条件不成立 (no)。

(3) 循环结构。循环结构的程序是指能够重复执行一段程序的结构，循环结构根据条件判断的结果，来决定是否重复执行一段程序，将重复执行的程序段称为循环体。图 1.5 中分别给出了当型循环和直到型循环的流程图，图中使用 S 表示循环体，使用 B 表示条件。

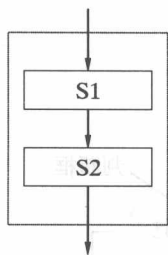


图 1.3 顺序结构流程图

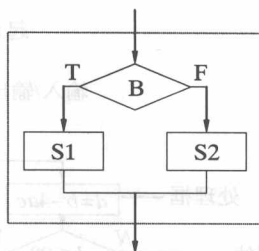
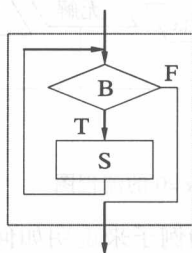
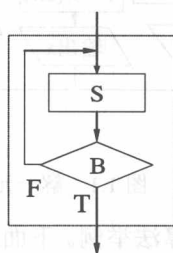


图 1.4 选择结构流程图



当型循环



直到型循环

图 1.5 循环结构流程图

2. 程序三种基本结构的 N-S 流程图表示

在图 1.6 中画出了顺序结构、选择结构、当型循环结构和直到型循环结构的传统流程图和对应的 N-S 流程图，可以通过对比来进行学习。其中的当型循环结构和直到型循环结构都属于三种基本结构中的循环结构。

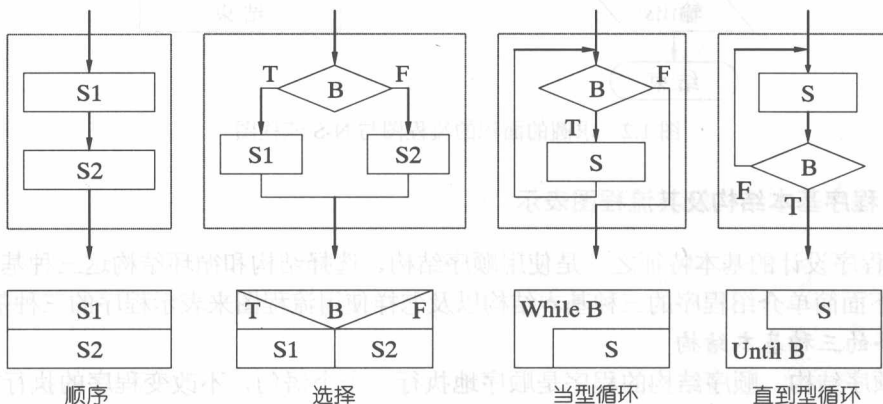


图 1.6 三种基本结构的传统流程图和对应的 N-S 流程图

需要指出的是，在 C 语言中，使用 `while` 语句可以构成当型循环，而使用 `do-while` 语句可以构成直到型循环。但是，使用 `do-while` 语句构成的直到型循环，与上述直到型循环稍有不同，其区别在于，使用 `do-while` 语句时，当条件 S 成立时，则循环，否则脱离循环。由于使用 `do-while` 语句构成的循环是先执行循环体，后判断条件是否成立，所以将它归类为直到型循环。使用 `do-while` 语句构成循环的流程图和对应的 N-S 流程图如图 1.7 所示。

例 1.2 顺序结构举例，求两数之和。其流程图和 N-S 流程图如图 1.8 所示。

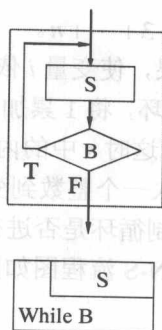


图 1.7 使用 do-while 语句构成循环的流程图和对应的 N-S 流程图

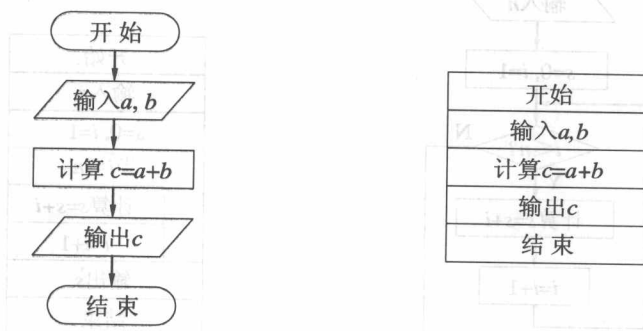


图 1.8 顺序结构的流程图与 N-S 流程图

算法：首先将被加数和加数分别输入到变量 a 和 b 中保存起来，然后计算变量 a 和 b 中的数据之和，并将求和结果存入变量 c ，最后输出变量 c 中的求和结果。

例 1.3 选择结构举例，解方程 $ax+b=0$ 。

算法：将方程的两个系数分别输入到变量 a 和 b ，然后判断系数 a 是否等于 0，如果 a 不等于 0，则计算方程的根，并将计算结果存入变量 x ，然后输出 x 的值，即输出方程的根。如果 a 等于 0，则判断系数 b 是否等于 0，如果等于 0，则输出“无穷多解”的信息，否则输出“无解”的信息。其流程图和 N-S 流程图如图 1.9 所示。

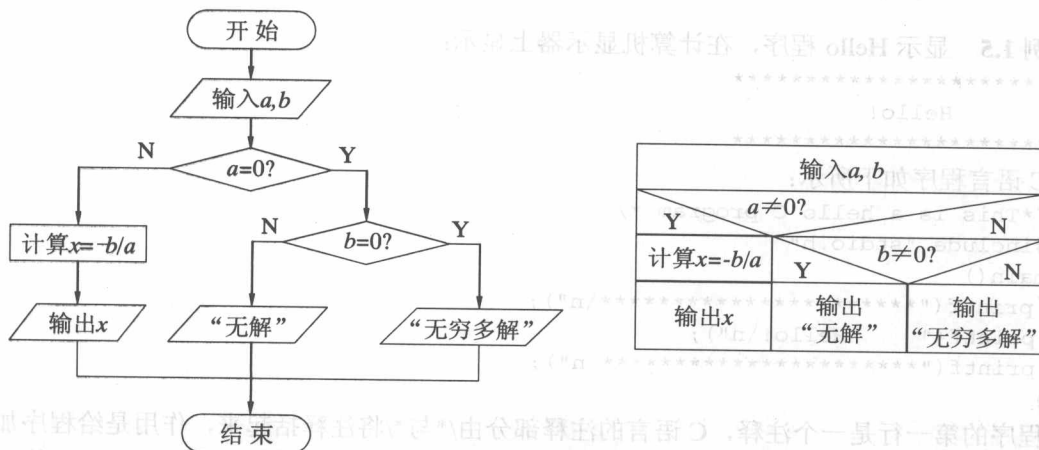


图 1.9 选择结构的流程图与 N-S 流程图

例 1.4 循环结构举例，求 $s=1+2+3+\cdots+n$ 。

设计思路：使用变量 s 存放累加结果，使变量 i 依次为 1、2、3、 $\cdots$ 、 n ，通过循环，分别将 1、2、3、 $\cdots$ 、 n 累加到 s 中，第 1 次循环，将 1 累加到 s ，第 2 次循环，将 2 累加到 s ， $\cdots$ ，依此类推，第 n 次循环，将 n 累加到 s ，这时 s 中的内容即为所求，通过循环控制条件终止循环，然后输出求和结果。流程：首先输入一个整数到变量 n ，确定求和项的终值，然后将 s 清 0， i 置 1，通过循环控制条件 $i \leq n$ 来控制循环是否进行，使得当 n 累加到 s 之后，结束循环，输出求和结果。其当型循环的流程图和 N-S 流程图如图 1.10 所示。

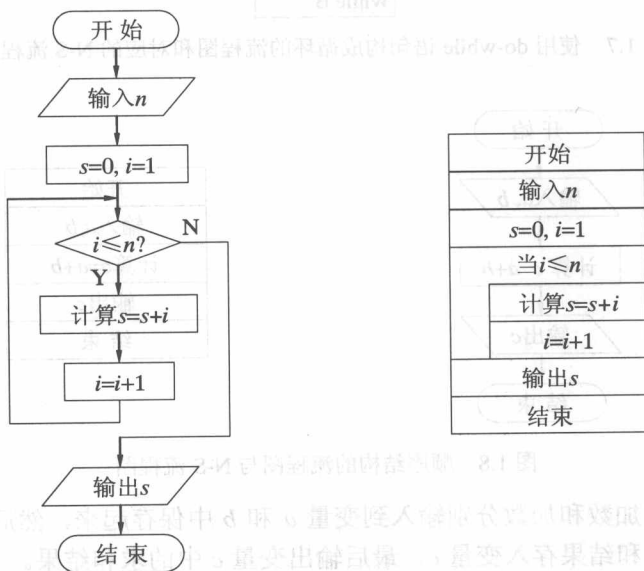


图 1.10 循环结构的流程图与 N-S 流程图

1.3 C 语言程序的基本构成

1.3.1 一个简单的 C 语言程序

例 1.5 显示 Hello 程序，在计算机显示器上显示：

Hello!

C 语言程序如下所示：

```

/*This is a hello C program.*/
#include "stdio.h"
main()
{printf("*****\n");
 printf("    Hello!\n");
 printf("*****\n");
}
  
```

程序的第一行是一个注释，C 语言的注释部分由 `/*` 与 `*/` 将注释括起来，作用是给程序加以注释，以帮助程序设计者阅读程序。注释部分只在程序清单中显示出来，并不被执行，换句话说，是程序的非执行部分。注释可以单独写在一行上，也可以写在某个语句之后。程序的第二