

高等职业教育计算机规划教材

C语言程序设计

李学军 主编

张东辉 余明艳 张桥珍 副主编

RETURN



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



高等职业教育计算机规划教材

C 语言程序设计

李学军 主编
张东辉 余明艳 张桥珍 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书采用理论讲授与上机实验相结合的方式,较全面系统地介绍了C语言的基础知识,C语言程序设计的基本方法和基本技巧,C语言程序设计的三种基本结构以及数组、函数、编译预处理、指针、结构、文件操作,C语言的综合应用等内容。

本书是作者根据多年的教学经验编写而成,在内容编排上尽量体现出易学的特点,讲解清晰细致,便于读者阅读理解;各章之后附有练习与思考题,针对性强;实验指导目的明确,可操作性强;综合实例深入浅出,通俗易懂。

本书适合作为高职高专院校各专业“C语言程序设计”课程的教学用书,也可作为全国计算机等级考试(二级C)的辅导教材,还可作为其他人员自学C语言的参考用书。

图书在版编目(CIP)数据

C语言程序设计/李学军主编. —北京:中国铁道出版社, 2008.6

高等职业教育计算机规划教材

ISBN 978-7-113-08474-5

I. C… II. 李… III. C语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2008)第094068号

书 名: C语言程序设计

作 者: 李学军 主编

策划编辑: 严晓舟 秦绪好

责任编辑: 王占清

编辑助理: 辛 杰 张 丹

封面设计: 付 巍

编辑部电话: (010) 63583215

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: 三河市华业印装厂

版 次: 2008年7月第1版 2008年7月第1次印刷

开 本: 787mm×1092mm 1/16 印张: 19.25 字数: 447千

印 数: 3 000册

书 号: ISBN 978-7-113-08474-5/TP·2649

定 价: 28.00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。

高等职业教育计算机规划教材

主 任：姚卿达

副主任：李 洛 贺 平 杨 利

徐人凤 王世杰

委 员：（按姓氏字母先后为序）

常理民 陈遵德 郭庚麒 蒋方纯

罗南林 王树勇 吴教育 肖石明

许龙飞 余少华 张国海 朱志辉

近几年,为适应区域和地方经济发展对技术应用型人才的需要,满足广大青年学生接受高等教育的愿望,国家颁布了大力发展职业教育的决定,在《国务院关于大力发展职业教育的决定》(国发[2005]35号)中提出了“以服务为宗旨,以就业为导向”的职业教育办学方针,并规定“大力推行工学结合、校企合作的培养模式”。教育部《关于全面提高高等职业教育教学质量的若干意见》(教高[2006]16号)中指出,“高等职业教育作为高等教育的一个类型,肩负着培养面向生产、建设、服务和管理第一线需要的高技能人才的使命”,规定“大力推行工学结合,突出实践能力培养,改革人才培养模式”。

按照国务院决定和教育部的意见,大力发展高等职业教育,推行工学结合和“2+1”教育模式,目前高职高专教育办学规模已占全国普通高等教育的一半以上,成为高等教育走向大众化的重要方面军,为了保证培养质量,很需要一套能适合工学结合教育模式的教材。

为贯彻国务院决定和教育部的意见,指导高职高专院校以就业为导向,深化高等职业教育改革,改进人才培养模式和教学工作,加强学生实践操作技能的训练,促进高职高专教育持续、健康和协调发展,计算机学会高职高专分会为高职高专院校提供了一个相互交流、相互促进学习的平台,共同探讨高职高专教育的教学模式、实践模式、专业建设、课程建设、教材建设等发展问题。在教材建设方面,计算机学会高职高专分会与中国铁道出版社合作,共同策划组织了本套“高等职业教育计算机规划教材”的编写。

本套教材面向工学结合教学、面向职业技能、面向“双证书”。

(1) 本套教材根据高等职业教育的教学规律和特点,借鉴国外成功的职业教育教材经验,运用现代教育理念编写。

(2) 编写本套教材的目标是培养面向生产、建设、管理、服务第一线需要的思想素质高、实践技能强、具有良好职业道德的高技能人才。

(3) 本套教材依照国家职业分类标准调整教材内容,兼顾证书课程考试大纲与专业教学大纲,改进了人才培养方案,创新了人才培养模式,强化了学生技能训练,使学生在获得学历证书的同时顺利获得相应的职业资格证书,增强就业竞争能力。

本套教材按照职业资格标准的要求,强调以培养学生的动手能力、实践能力和可持续发展能力为本,理论知识以应用为目的,理论和实践完全贯通,实现了“理论与实践”的有机融合。

本套教材由高职高专院校具有多年教学经验的老师设计教材结构框架,组织第一线教学的专业老师和企业兼职教师编写,最后经全国高职高专计算机教育专家审定,由专家和出版社进行质量把关。

在此,向为本套教材设计、编写和审定付出辛勤劳动的各位同仁,表示衷心的感谢!

编审委员会主任: 

C 语言是一种结构化、模块化、可编译的通用程序设计语言，它具有功能强、使用方便灵活、程序移植性好等特点，并兼备高级语言和低级语言的许多优点，已发展成为应用最广泛的计算机程序设计语言之一。高等院校几乎所有的计算机专业和许多理工科专业都开设了“C 语言程序设计”课程。为了给 C 语言的初学者和讲授 C 语言程序设计课程的教师们提供一本简明易懂、又易学的 C 语言程序设计基础教材，我们编写了本书。

本书根据高职高专院校各专业“C 语言程序设计”课程的要求和“全国计算机等级考试（二级 C）考试大纲”的要求，以技术应用型人才为培养目标，以 C 语言程序设计基础知识和应用能力的培养为主要内容，重点介绍了 C 语言的基础知识、C 语言模块化程序设计的基本方法和基本技巧、C 语言程序设计的三种基本结构以及数组、函数、编译预处理、指针、结构、文件操作、C 语言的综合应用等内容。每章由学习目标、教学指引、正文、练习与思考、实验五部分组成，信息量大，图文并茂，讲解清晰细致，深入浅出，针对性强，通俗易懂且例题丰富，所有例题均在 Turbo C++3.0 版本的编译系统下通过运行，并配套有 C 语言程序设计基础教程课件和源代码，供读者下载（<http://edu.tqbooks.net>）。

本书是作者根据多年的教学经验编写而成，在内容编排上体现出易学的特点，便于读者阅读理解；适合作为高职、高专院校各专业“C 语言程序设计”课程的教学用书，也可作为全国计算机等级考试（二级 C）的辅导教材，同时也可作为其他人员自学 C 语言的参考用书。

本书由李学军任主编，张东辉、余明艳、张桥珍任副主编。全书共分 13 章，其中第 2 章至第 6 章由李学军编写，第 1、7、8 章由张东辉编写，第 9、10 章由余明艳编写，第 11 章至第 13 章由张桥珍编写。全书由李学军负责统稿和定稿。

朱小平副教授、邱炳城副教授以及曾志红、张玉梅等同志对本书的内容安排提出了宝贵的意见，在此表示诚挚的感谢；另外，在编写过程中，还得到了不少同行和朋友们的关心、帮助，他们建设性的意见使本书增色不少，在此一并表示感谢。

由于作者水平有限，书中难免出现差错和疏漏，敬请读者赐教与指正，多提宝贵意见和建议，在此表示真挚的谢意，我们会在适当的时间进行修订和补充。

编 者

2008 年 5 月

第 1 章 C 语言概述	1
1.1 C 语言的发展简史和特点	1
1.1.1 C 语言的产生与发展	2
1.1.2 C 语言的特点	2
1.2 C 程序的结构与书写规则	2
1.2.1 C 程序的基本结构	2
1.2.2 如何编写可读性良好的 C 程序	4
1.3 C 语言的语句、标识符和关键字	4
1.3.1 C 语言的语句	4
1.3.2 C 语言标识符	4
1.3.3 C 语言关键字	5
1.4 C 程序的运行与调试	5
1.4.1 C 程序设计的基本步骤	5
1.4.2 如何使用 Turbo C++ 3.0 运行与调试 C 程序	6
本章小结	12
练习与思考	12
实验一 C 程序的运行环境和运行方法	13
第 2 章 基本数据类型、运算符与表达式	15
2.1 C 语言的数据类型	16
2.2 常量与变量	16
2.2.1 常量	16
2.2.2 变量	17
2.3 整型数据	18
2.3.1 整型常量	19
2.3.2 整型变量	19
2.4 实型数据	20
2.4.1 实型常量	20
2.4.2 实型变量	21
2.5 字符型数据	21
2.5.1 字符常量	21
2.5.2 字符变量	22
2.5.3 字符串常量	23
2.6 各种类型数据之间的混合运算	24
2.7 运算符及其表达式	25
2.7.1 算术运算符和算术表达式	26
2.7.2 自增与自减运算符	27

2.7.3 赋值运算符和赋值表达式	28
2.7.4 关系运算符和关系表达式	30
2.7.5 逻辑运算符和逻辑表达式	31
2.7.6 逗号运算符和逗号表达式	33
2.7.7 位运算符及其表达式	33
2.8 运算符的优先级及其结合性	36
本章小结	37
练习与思考	38
实验二 基本数据类型、运算符及表达式	43
第 3 章 C 语句和简单的 C 程序设计	45
3.1 C 语句概述	45
3.2 程序的三种基本结构	47
3.3 赋值语句	49
3.4 数据输出	49
3.4.1 printf()— 格式化输出函数	49
3.4.2 putchar()— 单个字符的输出函数	54
3.4.3 scanf()— 格式化输入函数	55
3.4.4 getchar()— 单个字符输入函数	57
3.5 顺序结构的 C 程序设计举例	57
本章小结	59
练习与思考	59
实验三 简单的 C 程序设计	63
第 4 章 选择结构程序设计	65
4.1 if 语句	65
4.1.1 if 语句的一般格式	65
4.1.2 if 语句的三种形式	66
4.1.3 if 语句的嵌套	69
4.1.4 条件运算符及条件表达式	70
4.2 switch 语句	72
4.3 选择结构的 C 程序设计举例	75
本章小结	79
练习与思考	79
实验四 选择结构程序设计	83
第 5 章 循环结构的 C 程序设计	85
5.1 循环语句	85
5.1.1 goto 语句	86
5.1.2 while 语句	88
5.1.3 do... while 语句	89

5.1.4 for 语句.....	91
5.2 break 语句和 continue 语句.....	94
5.2.1 break 语句.....	94
5.2.2 continue 语句.....	94
5.2.3 break 语句和 continue 语句的区别和联系.....	95
5.3 循环的嵌套.....	96
5.4 几种循环的比较.....	99
5.5 循环结构的 C 程序设计举例.....	100
本章小结.....	103
练习与思考.....	104
实验五 循环结构程序设计.....	109
第 6 章 数组和串	111
6.1 一维数组的定义和引用.....	112
6.1.1 一维数组的定义.....	112
6.1.2 一维数组元素的引用.....	112
6.1.3 一维数组元素的初始化.....	113
6.1.4 一维数组应用举例.....	114
6.2 二维数组的定义和引用.....	116
6.2.1 二维数组的定义.....	116
6.2.2 二维数组元素的引用.....	117
6.2.3 二维数组元素的初始化.....	118
6.2.4 二维数组应用举例.....	119
6.3 字符数组与字符串.....	121
6.3.1 字符数组的定义.....	121
6.3.2 字符数组的初始化.....	121
6.3.3 字符数组的引用.....	122
6.3.4 字符串.....	124
6.3.5 常用的字符串处理函数.....	124
6.4 数组应用举例.....	126
本章小结.....	130
练习与思考.....	131
实验六 数组与字符串的应用.....	134
第 7 章 函数	136
7.1 C 程序的模块化结构.....	136
7.1.1 概述.....	136
7.1.2 函数的种类.....	137
7.2 库函数.....	138
7.2.1 C 语言常用库函数.....	138
7.2.2 标准库函数的调用.....	139

7.3	函数的定义	139
7.3.1	函数的定义格式	139
7.3.2	函数的返回值与函数类型	141
7.3.3	对被调用函数的说明和函数原型	141
7.3.4	函数的形参与实参	143
7.4	函数的调用	144
7.4.1	函数的简单调用	144
7.4.2	函数的嵌套调用	145
7.4.3	函数的递归调用	146
7.5	数组作为函数参数	148
7.5.1	数组元素作为函数参数	148
7.5.2	数组名作为函数的形参和实参	149
7.6	由多个函数组成的 C 程序设计应用举例	151
	本章小结	152
	练习与思考	153
	实验七 函数	157
第 8 章	编译预处理	158
8.1	宏定义与符号常量	159
8.1.1	无参数的宏定义	159
8.1.2	带参数的宏定义	160
8.2	文件包含	162
8.3	条件编译	163
	本章小结	165
	练习与思考	165
	实验八 编译预处理	167
第 9 章	指针	168
9.1	什么是指针	168
9.2	指针和指针变量	169
9.2.1	指针变量的定义	169
9.2.2	指针变量的初始化	170
9.2.3	指针变量的引用	170
9.2.4	指针变量作函数参数	172
9.3	数组指针	174
9.3.1	指向数组元素的指针变量	174
9.3.2	通过指针引用数组元素	174
9.3.3	数组名作函数参数	177
9.4	字符串指针和指向字符串的指针变量	178
9.4.1	字符串的指针表示和引用	178
9.4.2	字符串指针作函数参数	180

9.5 函数指针	181
9.5.1 指向函数的指针变量	181
9.5.2 用函数指针调用函数	181
9.5.3 用指向函数的指针作函数参数	182
9.6 返回指针值的函数	183
9.7 指针数组和指向指针的指针	184
9.7.1 指针数组	184
9.7.2 指针数组作 main()函数的形参	185
9.7.3 指向指针的指针	186
9.8 指针应用举例	187
本章小结	192
练习与思考	193
实验九 指针的应用	196
第 10 章 结构体、共用体、枚举体和用户自定义类型	197
10.1 结构体	197
10.1.1 结构体的类型定义	197
10.1.2 结构体变量的定义	198
10.1.3 结构体变量的引用与初始化	199
10.1.4 结构体数组	201
10.1.5 指向结构体类型数据的指针	202
10.2 共用体	203
10.2.1 共用体类型的定义	203
10.2.2 共用体类型变量定义	203
10.2.3 共用体变量的引用	203
10.3 枚举体	205
10.4 用户自定义类型	207
10.5 应用举例	208
本章小结	215
练习与思考	216
实验十 结构体和共用体	221
第 11 章 C 语言的模块化程序设计	222
11.1 程序设计的基本方法	222
11.2 C 语言对模块化程序设计的支持	224
11.2.1 内部变量、外部变量及其作用范围	224
11.2.2 变量的存储方式	226
11.2.3 内部函数和外部函数	230
11.3 多个源程序文件组成的 C 程序设计案例分析	232
本章小结	234
练习与思考	234
实验十一 C 语言的模块化程序设计	239

第 12 章 文件	240
12.1 C 语言文件概述	240
12.1.1 文件的概念与分类	240
12.1.2 缓冲文件系统与非缓冲文件系统	241
12.1.3 文件的操作流程	242
12.2 文件的打开与关闭	242
12.2.1 文件的打开函数 <code>fopen()</code>	243
12.2.2 文件的关闭函数 <code>fclose()</code>	244
12.3 文件的读写	244
12.3.1 读 / 写一个字符函数	245
12.3.2 读 / 写一个字符串函数	247
12.3.3 读 / 写一个数据块函数	247
12.3.4 对文件进行格式化读 / 写函数	249
12.3.5 读 / 写函数的选用原则	250
12.4 位置指针与文件定位	250
12.4.1 随机读写与 <code>fseek()</code> 函数	250
12.4.2 返回文件当前位置的函数 <code>ftell()</code>	251
12.4.3 位置指针复位函数 <code>rewind()</code>	251
12.5 出错检测	252
12.5.1 <code>ferror()</code> 函数	252
12.5.2 <code>clearerr()</code> 函数	252
12.6 应用举例	252
本章小结	254
练习与思考	255
实验十二 文件	259
第 13 章 C 语言的综合应用	260
13.1 顺序表的插入与删除	260
13.2 利用结构体变量构成链表	263
13.3 绘制图形实例	269
13.4 综合应用案例分析	275
本章小结	278
练习与思考	278
实验十三 C 语言的综合应用	282
附录 A ASCII (美国信息交换标准编码) 表	283
附录 B C 语言的语法提要	284
附录 C 运算符的优先级与结合性	285
附录 D Turbo C 常用库函数	286
附录 E C 语言使用中的常见错误	290
参考文献	294

第 1 章 C 语言概述

📖 主要内容

- C 语言的发展简史及特点
- C 程序的结构及书写规则
- C 语言的语句、标识符和关键字
- C 程序的运行与调试

📖 核心知识点

- C 程序的基本结构
- C 程序的书写规则
- C 程序的编辑、编译、运行和调试的一般过程

📖 重点难点

- C 语言的基本结构
- Turbo C 编译系统工作环境参数的设置
- 编辑、编译、运行、调试 C 程序的完整的上机过程及 C 程序的编辑、调试技巧

📖 学习目标

- 了解 C 语言的发展历史和 C 语言的特点
- 掌握 C 程序的基本结构与书写规则
- 掌握 C 程序的编辑、编译、运行和调试的一般过程

1.1 C 语言的发展简史和特点

1.1.1 C 语言的产生与发展

C 语言是国际上广泛流行的、很有发展前途的计算机高级程序设计语言。它既适合作为系统描述语言，又可用于编写应用软件。

对 C 语言的研究起源于对系统程序设计的深入研究和发 展。1967 年，英国剑桥大学的 M.Richards 在 CPL 语言的基础上实现并推出了 BCPL (Basic Combined Programming Language)。1970 年，美国贝尔实验室的 Ken Thompson 以 BCPL 为基础，又做了进一步简化，设计出了 B 语言，并用 B 语言编写了第一个 UNIX 操作系统。由于 B 语言过于简单，功能有限，1972 年贝尔实验室的 Dennis M.Ritchie 在 B 语言的基础上设计出了 C 语言。1973 年，贝尔实验室的 K.Thompson 和 Dennis M.Ritchie 合作，首先用 C 语言重新改写了 UNIX 操作系统。此后，伴随

着 UNIX 操作系统的发展, C 语言越来越广泛地被人们接受和应用。

至此, C 语言不断得到改进, 但主要还是作为实验室产品在使用, 因为它仍然依赖于具体的机器。直到 1977 年才出现了独立于具体机器的 C 语言编译版本。1978 年, Brian W.Kernighan 和 Dennis M.Ritchie 正式出版了影响深远的《The C Programming Language》一书, 书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础, 它被称为标准 C 语言。

C 语言的标准化工作是从 20 世纪 80 年代初期开始的。1983 年, 美国国家标准化协会(ANSI)颁布了 C 语言的新标准 ANSI C。由于 C 语言的不断发展, 1987 年, 美国国家标准化协会又颁布了新标准—87 ANSI C。1990 年, 87 ANSI C 成为 ISO C 的标准。目前流行的 C 语言编译系统都是以它为基础的。

目前, 我国计算机上常用的 C 语言版本主要有 Borland International 公司的 Turbo C 和 Microsoft 公司的 Microsoft C 等。

1.1.2 C 语言的特点

C 语言是目前世界计算机领域广泛应用的一种高级程序设计语言, 具有较强的生命力。C 语言之所以能够存在且持续发展, 正是由于它自身具备的突出特点。

C 语言的主要特点如下:

(1) 语言简洁、使用方便。C 语言一共只有 32 个关键字, 程序书写形式自由, 与其他语言相比, C 语言的书写形式更为直观、精炼。

(2) 适应性强、应用范围广。C 语言能够适应从 8 位微型机到巨型机的所有机种, 并可应用于系统软件以及各个领域的应用软件。

(3) 运算符丰富, 语言的表达能力强。C 语言共有 34 种运算符, 它把括号、赋值号、强制类型转换等作为运算符处理, 使 C 语言运算类型丰富、灵活、多样, 功能强大。可直接处理字符, 访问内存地址, 进行位操作等。

(4) 数据结构系统化。C 语言具有现代化语言的各种数据结构, 且具有数据类型的构造能力。因此, 便于实现各种复杂的数据结构的运算。

(5) 具有结构化的控制语句。C 语言是结构化的程序设计语言, 提供了各种控制语句(如 if、while、for、switch 等语句), 对程序的逻辑结构提供了很好的基础。其程序结构清晰, 层次分明, 有利于采用自顶向下、逐步求精的程序设计方法。

(6) 运行程序质量高, 程序运行效率高。试验表明, C 源程序生成的运行程序的效率仅比汇编程序的效率低 10% ~ 20%。

(7) 可移植性好。C 编译程序基本上不做修改就能应用于各种型号的计算机和各种操作系统上。

1.2 C 程序的结构与书写规则

1.2.1 C 程序的基本结构

用 C 语言编写的程序称为 C 程序或 C 源程序。下面通过一个简单的 C 程序实例, 分析 C 程序的基本组成和结构。

【例 1.1】简单的 C 程序。

```

main()                                / * 主函数 * /
{
    int a,b,c;                        / * 声明部分, 定义 a, b, c 为整型变量 * /
    scanf("%d,%d",&a,&b);            / * 由键盘输入 a 和 b 的值 * /
    c=max(a,b);                      / * 调用 max 函数, 将得到的值赋给 c * /
    printf("max=%d",c);              / * 输出 c 的值 * /
}
int max(int x,int y)                 / * 定义 max 函数 * /
{
    int z;                          / * max 函数中定义变量 z 为整型 * /
    if(x>y) z=x;                    / * 如果 x>y, 将 x 赋值给 z, 否则将 y 赋值给 z * /
    else z=y;                      / * 将 z 的值返回到函数的调用处 * /
    return(z);
}

```

这个程序的执行结果如下:

```

3,5 ✓
max=5

```

说明:

- (1) 本程序包含两个函数: main 主函数和 max 被调用的函数。
- (2) 由一对大括弧{}括起来的是函数体。
- (3) 在 main 函数中通过赋值语句“c=max(a,b);”调用 max 函数。
- (4) max 函数的功能是将 x 和 y 中较大者的值赋给变量 z。return 语句将 z 的值返回给主调函数 main。

由此可见, 一个完整的 C 程序结构有以下特点:

(1) C 语言是由函数构成的。函数是 C 语言程序的基本单位。其中 main 函数是一个特殊的函数, 一个完整的 C 程序必须有且仅有一个 main 函数, 它是程序启动时的唯一入口(不管 main 函数的位置如何)。除 main 函数之外, C 程序也可包含若干个其他 C 标准函数和用户自定义的函数, 它们可以相互调用, 最终返回主函数结束程序。

(2) 一个函数是由函数说明和函数体两个部分组成。

① 函数的说明部分。确定函数名并说明函数名、函数类型、形式参数名及函数类型。例如 1.1 中:

int	max	(int x	,	int y)
↓	↓	↓		↓
类型名	函数名	形参 x 为整型		形参 y 为整型

② 函数体。由函数说明部分下的第 1 对大括弧{}内的一系列语句和注释构成。语句又包括变量的定义和可执行语句两部分。

变量定义: 如 main 函数中的“int a,b,c;”语句, max 函数中的“int z;”语句。

可执行语句: 完成当前函数功能的语句, 如“scanf("%d,%d",&a,&b);”语句等。

(3) 程序书写格式较自由, 一行可写几条语句, 一条语句也可以分行书写。

(4) 语句以分号结束, 例如“z=y;”语句。

(5) 在每条语句后, 可用/* */ 对该语句进行注释, 以增加程序的可读性。注释不影响语句的功能。

1.2.2 如何编写可读性良好的 C 程序

可读性良好的 C 程序可以使用结构化程序设计的方法来编写。这种程序不但便于编写、阅读、修改和维护,而且减少了程序出错的机会,提高了程序的可靠性,保证了程序的质量。

结构化程序设计强调程序设计风格和程序结构的规范化,提倡清晰的结构。怎样才能得到一个结构化的程序呢?如果我们面临一个复杂的问题,是难以一下子写出一个层次分明、结构清晰、算法正确的程序的。结构化程序设计方法的基本思路是,把一个复杂问题的求解过程分阶段进行,每个阶段处理的问题都控制在人们容易理解和处理的范围内。

具体来说,采取以下方法保证得到结构化的程序。

(1) 自顶向下;(2) 逐步细化;(3) 模块化设计;(4) 结构化编码。

在接受一个任务后应该怎样着手进行呢?有两种不同的方法:一种是自顶向下,逐步细化;一种是自下而上,逐步积累。以写文章为例来说明这个问题,有的人胸怀全局,先设想整篇文章分成哪几个部分,然后再进一步考虑每一部分分成哪几节,每一节分成哪几段,每一段应包含什么内容。用这种方法逐步分解,直到作者认为可以直接将各小段表达为文字语句为止。这种方法就叫做“自顶向下,逐步细化”。

另有些人写文章时不拟提纲,如同写信一样提起笔就写,想到那里就写到那里,直到他认为把想写的内容都写出来了为止。这种方法叫做“自下而上,逐步积累”。

显然,用第一种方法考虑周全,结构清晰,层次分明。作者容易写,读者也容易看。如果发现某一段内容不妥,需要修改,只需找出该部分,修改有关段落即可,与其他部分无关。我们提倡用这种方法设计程序。这就是用工程设计的方法来设计程序。

我们应当掌握这种“自顶向下,逐步细化”的设计方法。这种方法的过程是将问题求解由抽象到逐步具体化的过程。用这种方法便于验证算法的正确性,在向下一层展开之前应仔细检查本层设计是否正确,只有上一层正确的情况下才能向下细化。如果每一层设计都没有问题,则整个算法就是正确的。由于每一层向下细化时都不太复杂,因此容易保证整个算法的正确性。检查时也是由上而下逐步检查,这样做,思路清楚,有条不紊地一步一步进行,既严谨又方便。

结构化程序有三种基本结构:顺序结构、选择结构、循环结构。使用这三种基本结构编写的程序称为结构化程序。我们会在第 3 章、第 4 章、第 5 章中对这三种结构进行详细的讲解。

1.3 C 语言的语句、标识符和关键字

1.3.1 C 语言的语句

C 语言的语句是由关键字、标识符、运算符、常量和变量来构成的。

1.3.2 C 语言标识符

在程序设计中,常用具有一定意义的名字来标识程序中的变量、函数、数组、类、标号以及用户自定义的数据类型,以方便在程序设计中按名字来访问数据,这个名字称为标识符。

在 C 语言程序中,标识符由 1~8 个字符组成。构成标识符的字符只能是下画线“_”、数字(0~9)和英文字母,并且标识符的第一个字符只能是下画线或英文字母。

下列这些都是合法标识符：

address、Tel、name、password、var1、year_1999、_123、ID_view

下列这些都是非法标识符：

T-3、2abc、Tel#、[str]、mail@sina、+point、!key

C语言是一种对字母大小写敏感的语言，定义标识符必须注意字母的大小写，例如：
passWord和password在C语言中是两个不同的标识符。

1.3.3 C语言关键字

在C语言程序中，还有一类特殊的标识符，它们是不能被用户程序自定义的标识符，它们仅供系统专用。不能用作用户定义的标识符，称为关键字，又称系统保留字。

关键字是有特殊含义的系统标识符，它们不能随便被用作用户程序中定义的标识符。

C语言保留的关键字如下：

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

1.4 C程序的运行与调试

1.4.1 C程序设计的基本步骤

程序设计就是用某种程序语言编写程序的过程。

一个简单的程序设计一般包含以下四个步骤。

(1) 分析问题，建立数学模型

使用计算机解决具体问题时，首先要对问题进行充分的分析，确定解决问题的步骤。针对所要解决的问题，找出已知的数据和条件，确定所需的输入、处理及输出对象。将解题过程归纳为一系列的数学表达式，建立各种量之间的关系，即建立起解决问题的数学模型。

(2) 确定数据结构和算法

根据建立的数学模型，规划输入的数据和预期输出的数据，确定存放数据的数据结构，并选择合适的算法加以实现。算法是指为解决某一特定问题而采取的确定有限的步骤。对同一问题，每一个人的算法都不可能完全相同。

(3) 编写程序

根据确定的数据结构和算法，用自己所使用的程序设计语言把这个问题的解决方案严格地描述出来，就是编写程序。

(4) 调试程序

程序开发人员编写的程序称为源程序或源代码，源代码不能直接被计算机执行。源代码要经过编译程序编译，生成目标程序，然后链接其他相应的代码，最后生成可被计算机执行的可执行文件。在程序调试过程中，要不断分析所得到的运行结果，进行程序的测试及调整，直至获得预期的结果。