



新世纪

新世纪中等职业教育
计算机基础教育系列规划教材

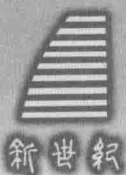
C 程序设计与训练

C CHENGXU SHEJI YU XUNLIAN

主编 李秉璋 李红卫



大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

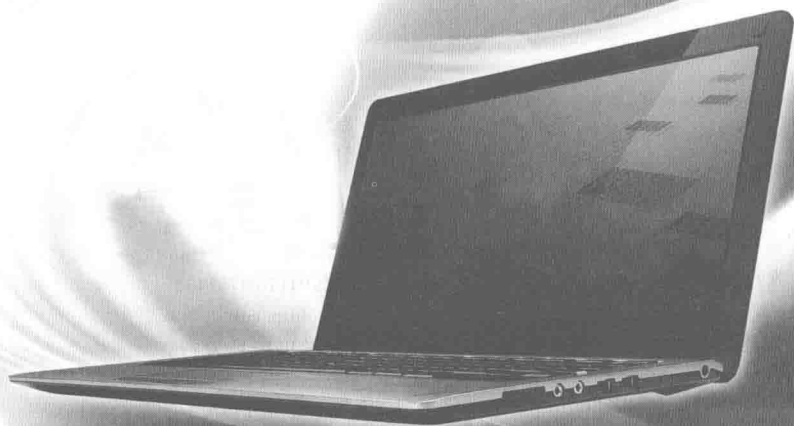


新世纪中等职业教育
计算机基础教育系列规划教材

C程序设计与训练

C CHENGXU SHEJI YU XUNLIAN

主编 李秉璋 李红卫



大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

图书在版编目(CIP)数据

C 程序设计与训练 / 李秉璋, 李红卫主编. —大连:
大连理工大学出版社, 2011. 8

新世纪中等职业教育计算机基础教育系列规划教材
ISBN 978-7-5611-6350-4

I. ①C… II. ①李… ②李… III. ①

C 语言—程序设计—中等专业学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 141713 号

主 编 李秉璋 李红卫

大连理工大学出版社出版

地址:大连市软件园路 80 号 邮政编码:116023

发行:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail:dutp@dutp.cn URL:http://www.dutp.cn

大连力佳印务有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm 印张:21.75 字数:495 千字

印数:1~2000

2011 年 8 月第 1 版

2011 年 8 月第 1 次印刷

责任编辑:潘弘喆

责任校对:潘素君

封面设计:张莹

ISBN 978-7-5611-6350-4

定 价:40.00 元

大连理工大学出版社
Dalian University of Technology Press
地址:大连市软件园路 80 号 邮编:116023
电话:0411-84708842 传真:0411-84701466
E-mail:dutp@dutp.cn URL:http://www.dutp.cn

前言

C语言是一种面向过程的计算机程序设计语言,它是目前众多计算机语言中举世公认的优秀结构化程序设计语言之一。它既具有高级语言的特点,又具有汇编语言的特点。它既可以作为开发系统软件的程序设计语言,也可以作为开发应用软件的程序设计语言。因此,它的应用范围十分广泛。

中等职业教育是我国职业教育的重要组成部分。中等职业教育培养目标定位于“综合职业能力强,在生产、服务、技术和管理第一线工作的高素质的劳动者和初中级专门人才”。为了适应中等职业教育计算机语言课程教学改革,培养更多掌握计算机基本理论、基本技能的人才,并使其中的佼佼者能够进入高一级的学校深造,依据教育部职业教育与成人教育司制订的《中等职业学校计算机及应用专业教学指导方案》、劳动部职业技能鉴定中心制订的《全国计算机高新技术考试技能培训和鉴定标准》组织编写了本教材。在编写过程中,充分研究了其他中职教材的特点、优点,兼顾了中职学生就业、升学、考级等不同层次的需要。

作 者
王 伟 等

本教材编写的原则是:突出职业素质教育和C语言应用能力培养,强调理论与实践结合,切实提高程序设计能力。教材采用由浅入深、循序渐进的方法讲授语言和程序设计方法。力求取材得当、通俗易懂、结构清晰、层次分明、易于理解。每个知识点或语法,都列举1~2个例题,并根据例题的难易程度,配有详略得当的例题分析、算法讲解、流程图、示意图、运行结果图,从而降低学习难度。作为一章的总结,每章安排“举例”一节,设计了若干个综合本章主要知识点,题意广泛的例题,并加以详细分析。作为一本偏重实践的教材,本书安排了20套程序设计训练题,既可作为程序设计上机练习题,也可作为参加相关考试的训练题。希望通过理论教学、课内实验、课后练习、作业、上机训练等一系列教学实践过程,切实提高学生的C程序设计能力。本书最后的附录则为学生学习C语言提供方便,特别是附录G“常见错误分析和程序调试”,能提高学生分析程序、排



新世纪

除错误的能力。

全书由两大篇章组成,上篇是程序设计部分,共分9章:第1章 C 程序设计概述;第2章数据类型与表达式;第3章基本语句与顺序结构程序设计;第4章选择结构程序设计;第5章循环结构程序设计;第6章数组及字符串;第7章函数;第8章文件;第9章自定义类型。每章结束安排有适量的涵盖该章主要知识点和技术的课后书面作业和上机实验题。下篇是程序设计训练部分,安排了20套程序设计训练题,并附有参考答案。建议将本教材分两个学期讲授。书中带“*”的章节的内容是否讲授,学校可以自行安排。

本教材由李秉璋、李红卫任主编。其中第1、6、7、8章及程序设计训练由李秉璋编写,第2、3、4、5、9章及附录由李红卫编写。本书所有例题、程序设计训练题均在 TC 3.0 和 VC++ 6.0 两个环境下调试通过。作者推荐使用 VC++ 6.0 作为学习本书、上机训练的开发环境。

本教材内容丰富、结构严谨、重点突出,主要面向学习 C 语言的初、中级用户。本书既可作为中等职业学校教材,也可作为 C 语言培训的教材,还可作为 C 程序设计人员的自学用书。

在本教材编写过程中,得到了江苏技术师范学院吴访升教授、徐小凤高级实验师及其他许多同行的帮助、指教,他们对本书的编写给出了许多宝贵意见。值本书出版之际,向他们表示衷心感谢。

诚恳希望广大读者在使用本教材过程中提出宝贵意见,以便我们在修订中不断改进和完善,使本教材成为中等职业教育的精品教材。

所有意见和建议请发往: dutpgz@163.com

欢迎访问我们的网站: <http://www.dutpgz.cn>

联系电话:0411-84707492 84706104

编者

2011年8月

目 录

上 篇 程序设计

第 1 章 C 程序设计概述	3
1.1 计算机语言和程序设计	3
1.1.1 计算机语言	3
1.1.2 程序与程序设计	5
1.2 算法的概念与表示方法	5
1.2.1 算法的概念	5
1.2.2 算法的表示	6
1.2.3 算法描述的三种基本结构	7
1.3 C 语言概述	9
1.3.1 C 语言的产生和发展	9
1.3.2 C 语言的特点	10
1.3.3 C 语言程序结构	10
1.3.4 C 程序的格式特点	13
1.4 C 程序的设计步骤与运行环境	13
1.4.1 C 程序的设计步骤	13
1.4.2 Visual C++ 6.0 集成开发环境	14
1.4.3 Turbo C 3.0 集成开发环境	18
本章小结	20
习 题	20
第 2 章 数据类型与表达式	21
2.1 C 语言基本字符集和词汇	21
2.1.1 C 语言基本字符集	21
2.1.2 词 汇	22
2.2 C 语言的数据类型	23
2.3 常 量	24
2.3.1 整型常量	24
2.3.2 实型常量	24
2.3.3 字符常量	25
2.3.4 字符串常量	26

2.3.5 符号常量	27
2.4 变 量	28
2.4.1 整型变量	29
2.4.2 实型变量	31
2.4.3 字符型变量	32
2.5 运算符和表达式	34
2.5.1 C 语言运算符与表达式简介	34
2.5.2 算术运算符和算术表达式	34
2.5.3 赋值运算符和赋值表达式	38
2.5.4 逗号运算符和逗号表达式	40
2.5.5 条件运算符和条件表达式	41
* 2.6 指针类型	42
2.6.1 指针的概念	42
2.6.2 指针变量的定义	43
* 2.7 位运算符	43
2.8 类型转换	45
2.8.1 自动类型转换	46
2.8.2 强制类型转换	46
本章小结	47
习 题	48
第 3 章 基本语句与顺序结构程序设计	50
3.1 C 语言的基本语句	50
3.2 数据输入与输出	51
3.2.1 数据输入输出的概念	51
3.2.2 字符数据的输入输出函数	51
3.2.3 格式化输入与输出函数	52
3.3 顺序结构程序设计	57
3.4 顺序结构程序设计应用举例	59
本章小结	63
习 题	63
第 4 章 选择结构程序设计	68
4.1 问题的提出	68
4.2 关系运算与逻辑运算	69
4.2.1 关系运算	69
4.2.2 逻辑运算	70
4.3 if 语句	73
4.3.1 if 语句的三种基本形式	73

4.3.2 if 语句的嵌套	76
4.4 switch 语句	81
4.5 选择结构程序设计应用举例	84
本章小结	88
习 题	88
第 5 章 循环结构程序设计	94
5.1 while 循环语句	94
5.2 do...while 循环语句	95
5.3 for 循环语句	97
5.4 循环结构的嵌套	100
5.5 转移语句	103
5.6 循环结构程序设计应用举例	107
本章小结	113
习 题	113
第 6 章 数组及字符串	118
6.1 一维数组的定义和引用	118
6.1.1 一维数组的定义及初始化	118
6.1.2 一维数组元素的引用	120
6.1.3 一维数组的应用——排序	122
6.1.4 一维数组的应用——查找	126
6.1.5 一维数组的其他应用举例	129
6.2 二维数组的定义和引用	131
6.2.1 二维数组的定义及初始化	131
6.2.2 二维数组元素的引用	132
6.2.3 二维数组应用举例	134
6.3 字符数组	137
6.3.1 字符数组的定义及初始化	137
6.3.2 字符数组的引用	138
6.3.3 字符串与字符数组	139
6.3.4 gets 函数和 puts 函数	143
6.3.5 字符串处理的常用系统函数	145
6.3.6 字符数组及字符串应用举例	147
6.4 数组应用举例	149
本章小结	152
习 题	152
第 7 章 函 数	158
7.1 函数的概念	158

7.2 函数的定义	160
7.2.1 函数定义的一般形式	160
7.2.2 函数的定义格式	160
7.3 函数调用	163
7.3.1 函数调用方式	163
7.3.2 函数的声明	165
7.3.3 函数的形式参数和实际参数	166
7.3.4 函数定义与调用举例	168
7.4 函数的嵌套调用	169
7.5 函数的递归调用	172
7.6 数组作为函数参数	174
7.7 局部变量、全局变量	177
7.7.1 局部变量	178
7.7.2 全局变量	179
* 7.8 变量的存储类别	180
7.8.1 变量的存储区域	180
7.8.2 变量的存储类型	180
7.8.3 变量生命期及其与存储类型、作用域的关系	184
7.9 函数应用举例	185
本章小结	189
习 题	190
* 第8章 文 件	198
8.1 C文件概述	198
8.1.1 文件分类	198
8.1.2 文件名	199
8.1.3 文件指针	200
8.2 文件的打开与关闭	200
8.2.1 文件的打开	200
8.2.2 文件的关闭	202
8.2.3 标准设备文件的打开与关闭	202
8.3 文件的读写	203
8.3.1 字符读写函数 fgetc 和 fputc	203
8.3.2 格式化读写函数 fscanf 和 fprintf	205
8.3.3 数据块读写函数 fread 和 fwrite	207
8.3.4 字符串读写函数 fgets 和 fputs	209
8.4 文件的随机读写	210
8.4.1 文件定位	210
8.4.2 文件的随机读写	211

8.5 文件检测函数	213
8.6 文件操作举例	213
本章小结	214
习 题	215
* 第 9 章 自定义类型	218
9.1 枚举类型	218
9.1.1 枚举类型的定义	218
9.1.2 枚举变量的使用	219
9.2 结构体类型	220
9.2.1 结构体类型的定义	220
9.2.2 结构体变量的定义和使用	221
9.3 共用体类型	223
9.4 类型定义符 typedef	224
9.5 自定义数据类型应用程序举例	225
本章小结	229
习 题	230

下 篇 程序设计训练

程序设计训练	235
程序设计训练一	235
程序设计训练二	238
程序设计训练三	241
程序设计训练四	243
程序设计训练五	247
程序设计训练六	250
程序设计训练七	253
程序设计训练八	256
程序设计训练九	259
程序设计训练十	262
程序设计训练十一	265
程序设计训练十二	268
程序设计训练十三	272
程序设计训练十四	274
程序设计训练十五	277
程序设计训练十六	280
程序设计训练十七	282

程序设计训练十八	285
程序设计训练十九	288
程序设计训练二十	290
参考答案	293
程序设计训练一答案	293
程序设计训练二答案	294
程序设计训练三答案	295
程序设计训练四答案	296
程序设计训练五答案	297
程序设计训练六答案	298
程序设计训练七答案	298
程序设计训练八答案	300
程序设计训练九答案	300
程序设计训练十答案	302
程序设计训练十一答案	303
程序设计训练十二答案	304
程序设计训练十三答案	305
程序设计训练十四答案	306
程序设计训练十五答案	307
程序设计训练十六答案	308
程序设计训练十七答案	309
程序设计训练十八答案	310
程序设计训练十九答案	311
程序设计训练二十答案	312
附 录	314
附录 A ASCII 码对照表	314
附录 B C 语言中的关键字	315
附录 C C 运算符、优先级和结合性	316
附录 D C 库文件	318
附录 E C 语言常用库函数	319
附录 F C 语言常用语法提要	324
附录 G 常见错误分析和程序调试	328
G.1 常见错误分析	328
G.2 错误检测与程序调试	334
参考文献	335

上
篇

程序设计

第1章

C 程序设计概述

现在,电子计算机已经渗透到了人类生产和生活的所有领域,成为人们认识世界改造世界的重要工具。而计算机的工作是由计算机程序来控制的,程序又是人们根据问题的要求、利用程序设计语言事先编写的。C 语言是一门非常优秀、应用广泛的结构化程序设计语言,它具有简洁、紧凑、灵活和可移植性强等优点,深受广大编程人员的喜爱,并得到广泛的应用。

本章对计算机语言、程序、算法等概念,对 C 语言发展、特点、C 程序的结构、C 程序设计等方面进行简单的介绍。最后介绍两种 C 程序开发环境的使用。

学习目标:

1. 了解程序、程序设计、程序设计语言的概念以及程序设计语言的分类。
2. 了解算法的概念、基本特征,掌握算法的常用表示方法。
3. 了解 C 语言程序的基本结构,掌握 C 语言程序设计的一般过程。
4. 基本掌握在 VC++ 6.0 或 TC 3.0 环境下编制、调试程序的步骤。

1.1 计算机语言和程序设计

1.1.1 计算机语言

要让计算机按人们的意图进行工作,就必须使计算机能理解和执行人们给它的指令。如同人与人交流要通过语言一样,人与计算机之间的交流也要通过特定的语言,这就是计算机语言。计算机语言是人机交流的媒介,需要人和计算机都能理解它。更为准确地说,计算机语言通常是一个能够完整、准确地表达人们的意图,并用于指挥或控制计算机工作的“符号系统”。

每一种语言都有各自的语法规则及其使用方法。而计算机程序则是为解决某一特定的问题采用某一种语言编写的指令序列。

计算机程序设计语言主要分为机器语言、汇编语言和高级语言。

1. 机器语言

到目前为止,所有计算机只能识别 0 和 1 这样的二进制信息。每一种类型的计算机

都分别规定了由若干位二进制信息组成一个指令。例如,某 8 位字长的计算机以 10000000 表示“加”操作,以 01110000 表示“赋值传送”操作。这种计算机能直接识别和执行的二进制形式的指令称为机器指令。每一条机器指令都是计算机设计者通过计算机的硬件设备赋予计算机的一个操作功能。每一种类型计算机都规定了一系列指令以实现各种不同的操作。某种计算机的指令集合即为该计算机的机器语言,或者说是计算机的指令系统。机器语言是依赖于具体计算机的,而非各类计算机通用的。不同类型的计算机机器语言是不同的。因此机器语言是面向机器的语言。

为解决某一问题,采用某种类型计算机的机器语言编写的一组机器指令,称为机器语言程序。机器语言程序能够被计算机直接识别和执行,效率高。但用机器语言编写程序,编程人员首先要熟记这类计算机的全部指令及指令的含义。编写程序时,程序员得自己处理每条指令和每一个数据的存储分配和输入输出,还得记住编程过程中每步所使用的工作单元处在何种状态。这是一件十分烦琐的工作。而且,编出的程序全是由 0 和 1 组成的代码,直观性差、易出错、不易维护、不可移植。因此除了极少数场合外,目前程序员已经不再去学习、使用机器语言了。

2. 汇编语言

为了克服机器语言程序难读、难编、难记、易出错、不可移植的缺点,人们就用与机器指令实际含义相近的英文缩写词、字母和数字等符号来取代机器指令。如用 ADD 表示“加”,用 SUB 表示“减”等。ADD、SUB 等称为“助记符”。这种以“助记符”代替二进制指令的语言称为汇编语言(又称符号语言)。汇编语言用符号代替了机器指令代码,而且助记符与指令代码一一对应,基本保留了机器语言的特点,因此它仍然是面向机器的语言。

用汇编语言编写的程序称为汇编语言源程序。由于使用了助记符号,汇编语言源程序并不能被计算机直接识别和执行,必须通过“汇编程序”的加工和翻译,生成能够被计算机直接识别和执行的二进制代码程序。经汇编程序翻译后的程序称为目标程序,目标程序是机器语言程序,它被安放在内存的预定位置上,就能被计算机直接执行了。

汇编语言与机器指令一样,仍然是面向机器的语言,使用起来还是比较烦琐费时,通用性也差。但是,用汇编语言编写的程序,其目标程序占用内存空间少,运行速度快,有着高级语言不可替代的作用。目前一些应用程序仍然使用汇编语言编写。

3. 高级语言

机器语言和汇编语言是面向机器的,可移植性差,并且难学、不易推广,人们习惯上将它们称为低级语言。随着计算机技术的发展,促使人们去寻求一些与人类自然语言相接近且能为计算机所接受的、语意确定、规则明确、自然直观、通用易学的计算机语言。这种与自然语言相近,能够为计算机接受并且执行的计算机语言称为高级语言。高级语言的出现,可以使一大批非计算机专业人员得心应手地使用计算机来实现其目的。

高级语言不再面向机器,而是面向过程。或者说,高级语言不必过多考虑计算机内部硬件构造、型号等差别,只要按照解题的过程用高级语言写出相应的程序,经过相应语言的编译程序生成计算机指令,计算机就能执行。由于高级语言接近人类语言,不同类型的

计算机中相同的高级语言基本没有差别,通用性强。因此高级语言已经成为程序设计的主流语言。采用高级语言编写的程序称为源程序。

1.1.2 程序与程序设计

所谓程序就是遵循一定语法规则组织起来、完成特定任务的代码或指令序列。而程序设计就是根据所完成的任务,设计解决问题的步骤,然后编写相应程序代码,并测试该代码的正确性,直到能够得到正确的运行结果为止的过程。通常程序设计应遵循一定的方法和原则,而不是个人随意编写。良好的程序设计风格是程序具备可靠性、可读性、可维护性的保证。在编写程序代码时,程序员必须按照一定的规范来描述问题的解决方案和步骤,这种规范就是程序设计语言。计算机程序设计语言具有一定的基本规则,固定的语法格式、特定的语义和使用环境,并且比自然语言要求更严格,不能出现二义性。

进行程序设计,一般包含以下几个步骤:

(1)明确需求:就是把问题陈述清楚,明确问题要处理的数据(即输入)和希望得到的结果(即输出),解决方案的约束条件和任何附加需求。

(2)确定数据结构和算法:确定程序中输入数据和输出结果的数据结构,选择合适的算法。例如,给定一个学生的数学、英语、语文三门课程的成绩(整数),求平均成绩,平均成绩要求保留一位小数。

要处理的数据(输入):三门课程的成绩,整数。

希望得到的结果(输出):平均成绩,小数。

根据问题的需求,可用公式计算:平均成绩=(数学+英语+语文)/3。

(3)编写程序:依据确定的数据结构和算法,用一种程序设计语言,按照规定的语法编写解决问题的步骤,即程序。

(4)测试程序:执行程序来测试它是否按预期工作,得到预期结果。如果没有得到预期结果,则需要分析,找出问题所在。

如果是需求分析问题,则需要返回到步骤(1),重新明确需求;

如果是数据结构和算法的问题,则需要返回到步骤(2),修改程序,重新确定数据结构和算法;

如果是编写程序的问题,则需要返回到步骤(3),修改程序,直至获得预期的结果。

(5)维护:通过修改程序来改正未检测到的错误。

1.2 算法的概念与表示方法

1.2.1 算法的概念

算法就是解决问题的方法(或思路)。对于同一个问题可以有不同的解题方法,也就是有不同的算法,一般应当选择简单、运算步骤少,运算快且占用内存开销小的算法。

算法分为数值计算算法和非数值计算算法。数值计算的目的是求数值解,例如求方程的根,求一个函数的定积分等。数值计算有现成的模型,可以运用数值分析方法,因此对数值计算算法的研究已经比较深入,算法也比较成熟。非数值计算是对大量数据进行加工处理,如合并、分类、排序、查找等,最常见的是用于事务处理领域,例如仓库库存管理、学籍档案管理等。

算法应具有以下几个特征:

- (1)可行性:指算法中的每一步都是计算机可以执行的,并能得到有效的结果。
- (2)确定性:指算法中的每一步必须有明确定义,不能有任何歧义。
- (3)有穷性:指算法必须在执行有限步骤后结束,而不能是无限的步骤,也不能陷入死循环。
- (4)可输入/输出信息:输入信息是算法加工的对象,而算法所解决问题的结果应当输出。大多数算法需要输入信息,输入可以从键盘输入的数据,也可以是其他方式传递给算法的数据。算法至少要有一个输出。

1.2.2 算法的表示

算法的表示方法有很多种,常用的算法表示方法有:自然语言,流程图,伪代码等。其中流程图和伪代码是程序设计人员常用的两种算法表示方法。

流程图是图形化的表示方法,用一些图框表示各种操作,用箭头表示算法流程,比较直观,易于理解。图 1-1 是常用流程图的符号。

(1)起止框:表示算法的开始或结束。一般内部只写“开始”或“结束”。

(2)输入/输出框:表示算法请求输入需要的数据或算法将某些结果输出。一般内部常常写“输入…”、“打印/显示…”。

(3)处理框:表示算法的某个处理步骤。

(4)判断框:对一个给定条件进行判断,根据判断结果决定如何执行其后的操作。判断框有一个入口,两个出口。

(5)流程线:指出了算法的执行方向。

(6)连接点:用于将画在不同地方的流程线连接起来。同一个编号的点是相互连接在一起的,实际上同一编号的连接点是同一个点,只是画不下才分开画。使用连接点,还可以避免流程线的交叉或过长,使流程图更加清晰。

(7)注释框:注释框不是流程图中必需的部分,不反映流程和操作,它只是对流程图中某些框的操作做必要的补充说明,以帮助人们更好地理解算法。

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。伪代码不用图形符号,没有严格的语法要求,书写方便,格式紧凑,便于向计算机语言过渡。

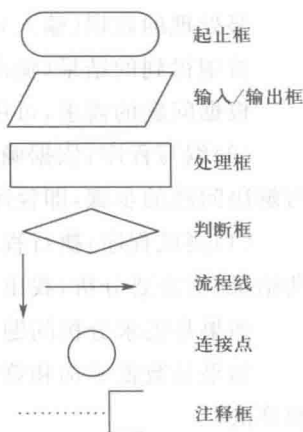


图 1-1 常用流程图的符号