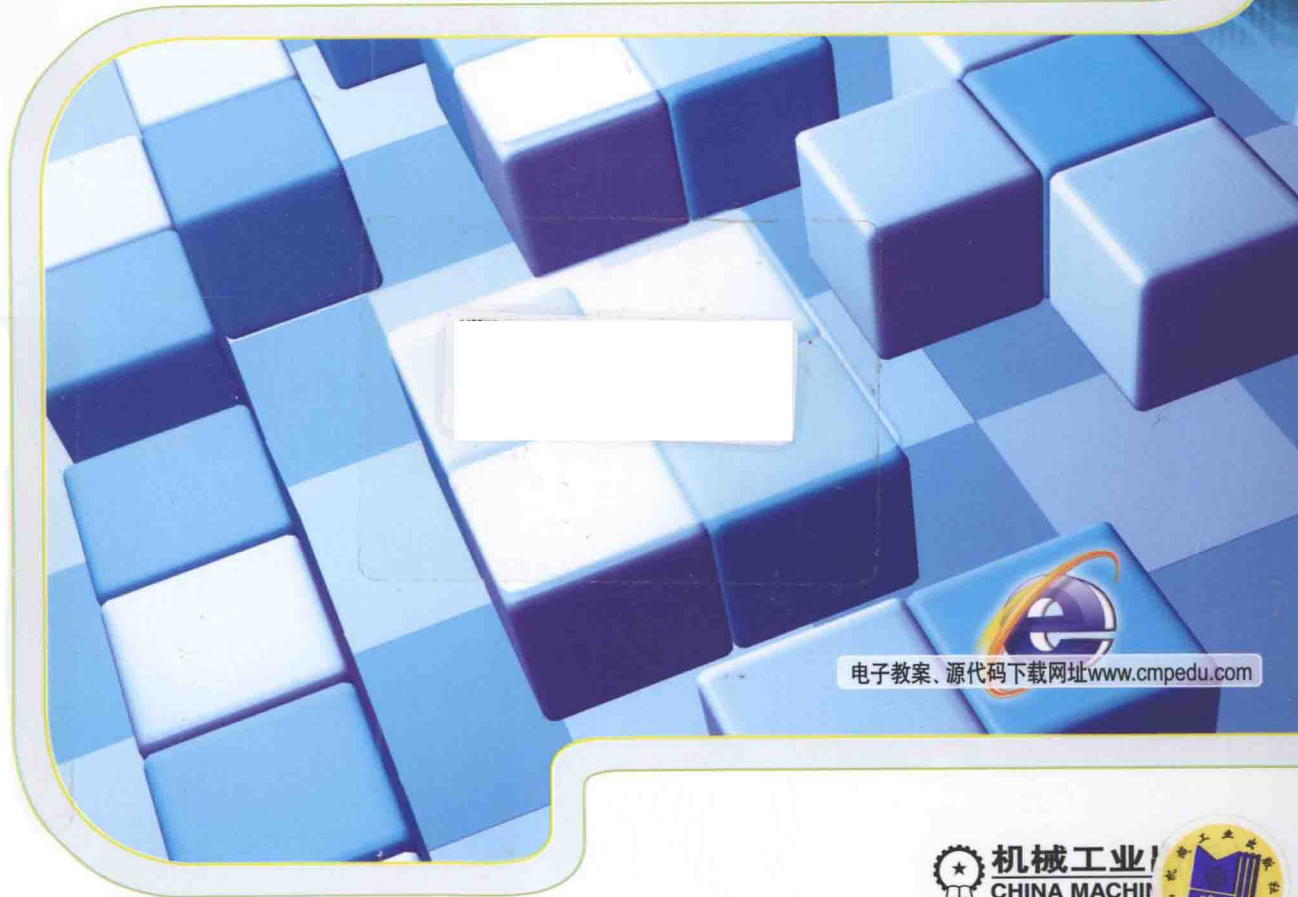




全国高等职业教育规划教材

C语言程序设计教程

杨丽波 朱 迅 汤承林 等编著



电子教案、源代码下载网址www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS



全国高等职业教育规划教材

C 语言程序设计教程

杨丽波 朱 迅 汤承林 等编著
刘长荣 主审



机械工业出版社

本书是根据高职高专学生的知识基础和学习特点、面向 C 语言程序设计初学者而编写的教程,书中从介绍 C 语言的每一个基础知识点入手,紧贴国家计算机等级考试的规范要求,理论与应用相结合,比较全面地介绍了 C 语言的基础知识和基本原理。

本书主要内容包括: C 语言概述、数据类型、运算符与表达式,顺序结构与基本输入/输出函数,选择结构,循环结构,数组,函数,指针,编译预处理,结构与共用,位运算和文件。

本书语言详实、图文采用适当,在保证一定广度和深度的前提下力求清晰易懂。各章节例题附有解题思路、参考代码、运行结果图示和注意事项等。每一章都精选了和全国计算机等级考试难度相匹配的习题帮助大家复习和巩固所学知识。

本书可作为各类高职高专、高等院校计算机专业及非计算机专业的课程教材,也可作为参加全国计算机等级考试和广大计算机爱好者学习 C 语言程序设计的参考书。

为配合教学,本书配有电子课件,读者可以登录机械工业出版社教材服务网 www.cmpedu.com 免费注册后下载,或联系编辑索取(QQ: 1239258369、电话(010) 88379739)。

图书在版编目(CIP)数据

C 语言程序设计教程/ 杨丽波等编著. —北京: 机械工业出版社, 2014. 2

全国高等职业教育规划教材

ISBN 978-7-111-45428-1

I. ① C… II. ① 杨… III. ① C 语言-程序设计-高等职业教育-教材
IV. ① TP312

中国版本图书馆 CIP 数据核字(2014)第 004501 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 鹿 征 曹帅鹏

责任印制: 张 楠

唐山丰电印务有限公司印刷

2014 年 2 月第 1 版·第 1 次印刷

184mm×260mm·13.75 印张·340 千字

0001-3000 册

标准书号: ISBN 978-7-111-45428-1

定价: 29.80 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

社服 务 中 心:(010)88361066

教 材 网:<http://www.cmpedu.com>

销 售 一 部:(010)68326294

机工官网:<http://www.cmpbook.com>

销 售 二 部:(010)88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010)88379203

封面无防伪标均为盗版

全国高等职业教育规划教材计算机专业

编委会成员名单

主 任 周智文

副 主 任 周岳山 林 东 王协瑞 张福强
陶书中 眭碧霞 龚小勇 王 泰
李宏达 赵佩华

委 员 (按姓氏笔画顺序)

马 伟	马林艺	万雅静	万 钢
卫振林	王兴宝	王德年	尹敬齐
史宝会	宁 蒙	安 进	刘本军
刘新强	刘瑞新	余先锋	张洪斌
张瑞英	李 强	何万里	杨 莉
杨 云	贺 平	赵国玲	赵增敏
赵海兰	钮文良	胡国胜	秦学礼
贾永江	徐立新	唐乾林	陶 洪
顾正刚	曹 毅	黄能耿	裴有柱

秘 书 长 胡毓坚

出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位,促进学生技能的培养,以及教材内容要紧紧密结合生产实际,并注意及时跟踪先进技术的发展等指导精神,机械工业出版社组织全国近 60 所高等职业院校的骨干教师对在 2001 年出版的“面向 21 世纪高职高专系列教材”进行了全面的修订和增补,并更名为“全国高等职业教育规划教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师,针对相关专业的课程设置,融合教学中的实践经验,同时吸收高等职业教育改革的成果而编写完成的,具有“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。在几年的教学实践中,本系列教材获得了较高的评价,并有多品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补过程中,除了保持原有特色外,针对课程的不同性质采取了不同的优化措施。其中,核心基础课的教材在保持扎实的理论基础的同时,增加实训和习题;实践性较强的课程强调理论与实训紧密结合;涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。同时,根据实际教学的需要对部分课程进行了整合。

归纳起来,本系列教材具有以下特点:

- 1) 围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。
- 2) 合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度,强调专业技术应用能力的训练,适当增加实训环节。
- 3) 符合高职学生的学习特点和认知规律。对基本理论和方法的论述要容易理解、清晰简洁,多用图表来表达信息;增加相关技术在生产中的应用实例,引导学生主动学习。
- 4) 教材内容紧随技术和经济的发展而更新,及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念,并积极支持新专业的教材建设。
- 5) 注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合,提高教学服务水平,为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快,加之我们的水平和经验有限,因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息,以利于我们今后不断提高教材的出版质量,为广大师生提供更多、更适用的教材。

机械工业出版社

前 言

C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。它功能强大、数据类型丰富、使用灵活，并具有良好的可移植性。许多大型应用软件都是使用 C 语言编写的，在对操作系统和对硬件操作的场合，使用 C 语言明显优于其他高级语言。可以说 C 语言兼有面向硬件编程的低级语言特性和可读性强的高级语言特性，因此，C 语言已经成为当前各高校学生一门基础性的必修课程。

本书主要面向各类高职高专、高等院校计算机专业和非计算机专业的学生，以及广大 C 语言程序设计零起点的读者。本书的主要特点在于内容完备、基础性强，在讲述各知识点时，先从基本原理入手，再结合实例，语言详实，易于理解。本书既可作为高职高专等学校 C 语言程序设计课程的教材，也是读者自学 C 语言的良好选择。

本书主要内容共 12 章，按照 C 语言的基础知识、基本理论由浅入深进行介绍。考虑到高职高专学生的特点和计算机等级考试的要求，本书介绍了 Visual C++ 6.0 环境的安装和调试方法，书中的实例都是在该环境中编译和执行的，具有很强的实用性。对于读者较难理解的指针、结构、链表和文件等内容，作者在编写时精心选择了大量既典型又与实际应用紧密结合的实例来帮助读者理解知识点。本书将位运算的内容单独作为一章，可以使读者更加系统地学习该内容，对今后编写设备驱动程序和嵌入式程序将会有很大帮助。

本书由淮安信息职业技术学院杨丽波、朱迅、汤承林和唐山职业技术学院吴东领共同编写。其中，杨丽波负责编写第 1、4、5、8、10 章，吴东领负责编写第 2、6、7 章，朱迅负责编写第 3、12 章，汤承林负责编写第 9、11 章。全书由杨丽波统稿，在本书编写过程中，得到了刘长荣、邢海霞等老师的大力支持和帮助，在此表示衷心的感谢！

由于作者水平有限，错误之处在所难免，恳请广大读者批评指正。

编 者

目 录

出版说明

前言

第1章 C语言概述	1
1.1 程序、算法和流程图	1
1.1.1 程序	1
1.1.2 算法	1
1.1.3 流程图	2
1.2 C语言的特点和源程序的 基本结构	3
1.2.1 C语言的特点	3
1.2.2 C语言程序的构成和 书写格式	4
1.2.3 C语言程序的3种基本结构	5
1.3 在Visual C++6.0环境开发 C语言程序	5
1.3.1 C语言程序的开发过程	5
1.3.2 Visual C++6.0的启动和 操作	6
1.3.3 C语言程序调试	9
本章小结	11
习题1	11
第2章 数据类型、运算符与表达式	13
2.1 C语言基本数据类型	13
2.2 常量与变量	13
2.2.1 常量和符号常量	13
2.2.2 变量	14
2.3 整型数据	15
2.3.1 整型常量	15
2.3.2 整型变量	15
2.4 实型数据	16
2.4.1 实型常量	16
2.4.2 实型变量	16
2.5 字符型数据	17
2.5.1 字符型常量	17
2.5.2 字符型变量	17

2.5.3 字符串常量	18
2.6 运算符与表达式	18
2.6.1 算术运算符与表达式	19
2.6.2 关系运算符与表达式	21
2.6.3 逻辑运算符与表达式	22
2.6.4 赋值运算符与表达式	23
2.6.5 条件运算符	24
2.6.6 逗号运算符	24
2.7 数据之间的混合运算与数据 类型转换	25
本章小结	26
习题2	27
第3章 顺序结构与基本输入/输出 函数	29
3.1 格式化输入/输出函数	29
3.1.1 格式化输入函数 scanf()	29
3.1.2 格式化输出函数 printf()	30
3.2 字符输入/输出函数	32
3.2.1 字符输入函数 getchar()	32
3.2.2 字符输出函数 putchar()	32
本章小结	32
习题3	33
第4章 选择结构	37
4.1 单分支 if 语句	37
4.2 二分支 if 语句	38
4.2.1 二分支 if 语句	38
4.2.2 二分支 if 语句与条件 运算符	39
4.2.3 if 语句嵌套	39
4.2.4 多分支 if 语句	40
4.3 switch 语句	42
4.4 综合实例	44
本章小结	47
习题4	48
第5章 循环结构	52

5.1 goto 语句	52	7.3.1 函数调用的一般形式	96
5.2 for 循环语句	53	7.3.2 函数的原型声明	96
5.3 while 循环和 do...while 循环 语句	55	7.3.3 函数的嵌套调用	98
5.3.1 while 循环语句	55	7.3.4 函数的递归调用	99
5.3.2 do...while 循环语句	56	7.4 数组作为函数参数	101
5.4 循环嵌套	58	7.4.1 数组元素作为函数实参 ...	101
5.5 break 语句和 continue 语句	60	7.4.2 数组名作为函数参数	102
5.5.1 break 语句	60	7.5 变量的存储类别、作用域和 生存期	103
5.5.2 continue 语句	61	7.5.1 局部变量和全局变量	103
5.6 综合实例	62	7.5.2 变量的存储类别	105
本章小结	65	7.6 内部函数和外部函数	107
习题 5	66	7.6.1 内部函数	107
第 6 章 数组	71	7.6.2 外部函数	107
6.1 一维数组	71	本章小结	108
6.1.1 一维数组的定义	71	习题 7	108
6.1.2 一维数组的初始化	72	第 8 章 指针	112
6.1.3 一维数组元素的引用	73	8.1 指针的概念和定义	112
6.2 二维数组	76	8.1.1 指针的概念	112
6.2.1 二维数组的定义	76	8.1.2 指针变量的定义和初始化 ...	112
6.2.2 二维数组元素的初始化	76	8.1.3 指针的引用	113
6.2.3 二维数组元素的引用	77	8.1.4 指向指针的指针	115
6.2.4 多维数组	78	8.1.5 指针数组	117
6.3 字符数组	79	8.2 指针变量与函数参数	118
6.3.1 字符数组的定义	79	8.3 指针与数组	120
6.3.2 字符数组的初始化	80	8.4 指针与二维数组	123
6.3.3 字符串与字符数组	80	8.4.1 二维数组的地址	123
6.3.4 字符串的输入输出	82	8.4.2 指向二维数组的指针变量 ...	125
6.3.5 字符串处理函数	83	8.5 指针与字符串	127
6.4 综合实例	86	8.5.1 用字符指针指向字符串 ...	127
本章小结	87	8.5.2 字符串指针作函数参数 ...	128
习题 6	88	8.6 指针与函数	129
第 7 章 函数	92	8.6.1 函数指针	129
7.1 函数的定义	93	8.6.2 返回值是指针的函数	130
7.2 函数的参数和返回值	94	8.7 带参数的 main() 函数	131
7.2.1 函数的形式参数和实际 参数	94	本章小结	132
7.2.2 函数的返回值	95	习题 8	133
7.3 函数的调用	96	第 9 章 编译预处理	135
		9.1 宏定义	135

9.1.1 不带参数的宏定义	135	11.1 位运算的基本概念	174
9.1.2 带参数的宏定义	137	11.2 位运算符	174
9.2 文件包含	138	11.2.1 位逻辑运算符	174
9.3 条件编译	139	11.2.2 移位运算	176
本章小结	142	11.2.3 位运算的应用实例	176
习题 9	142	本章小结	177
第 10 章 结构与共用	144	习题 11	177
10.1 结构的定义与结构变量的 操作	144	第 12 章 文件	178
10.1.1 结构的定义	144	12.1 文件概述	178
10.1.2 结构变量的定义	145	12.2 文件类型指针	179
10.1.3 结构变量的引用与赋值	147	12.3 文件的打开与关闭	179
10.1.4 结构变量的初始化	148	12.3.1 文件的打开 (fopen 函数)	179
10.2 结构数组	149	12.3.2 文件的关闭 (fclose 函数)	181
10.2.1 结构数组的定义和 初始化	149	12.4 文件的读写	181
10.2.2 结构数组的应用	150	12.4.1 字符读写函数 fgetc() 和 fputc()	181
10.3 结构与指针变量	151	12.4.2 字符串读写函数 fgets() 和 fputs()	183
10.3.1 指向结构变量的指针	151	12.4.3 二进制读写函数 fread() 和 fwrite()	184
10.3.2 指向结构数组的指针	153	12.4.4 格式化读写函数 fscanf() 和 fprintf()	185
10.3.3 指向结构变量的指针 作为函数参数	155	12.5 文件的随机读写	187
10.3.4 指向结构数组的指针作为 函数参数	156	12.5.1 文件的定位	187
10.4 动态内存分配	157	12.5.2 文件的随机读写	188
10.5 结构与链表	159	本章小结	189
10.5.1 链表概述	159	习题 12	189
10.5.2 链表操作	160	附录	192
10.6 共用	167	附录 A ASCII 码表	192
10.6.1 共用的概念	167	附录 B C 语言常用标准库函数	193
10.6.2 共用变量的引用	169	参考文献	212
本章小结	171		
习题 10	171		
第 11 章 位运算	174		

第1章 C语言概述

1.1 程序、算法和流程图

1.1.1 程序

“程序”是我们在日常生活中经常听到和用到的一个词，通常指完成某件事情的一系列动作和过程，例如我们去食堂吃饭通常要经过进入食堂、排队、打饭、刷卡、就餐、洗饭盆和离开食堂这样一系列的过程，我们必须按照顺序依次执行每个步骤，才能够完成这件事情，如果你因为上次饭盆未洗而先洗饭盆再排队的话，就属于特殊情况了。这就是日常生活中程序性活动的例子。

用计算机来完成某件事情的过程和在日常生活中解决某个问题的过程很相似，计算机提供给我们具有各种功能的指令，每一条指令对应着计算机能够执行的一个操作，这个操作可能是实现对数据的输出，也可能是实现对图形的绘制，还可能是实现对文字的拷贝等，我们可以将要实现的任务变成由这样的一系列指令组成的集合，通过在计算机中执行这组指令来完成这个任务。为了让计算机解决某个问题而编写的这组指令的集合就称为程序。

1.1.2 算法

“算法”从字面来讲似乎是指计算的方法，其实，不只是这样，在计算机中，为了解决一个问题而采取的方法和步骤，就称为“算法”，这个问题既可以是计算问题，也可以是图形处理问题，还可以是通过界面进行的数据查询等问题。或者说，算法就是对于各种问题所采用的方法的精确描述，而对于一个问题，通常可以找到多种解决方法，不同的方法解决问题的准确性和执行效率是不一样的，我们在选择算法的时候，要根据算法的准确性、可靠性和易理解性来找到最优化的算法。在用计算机来实现算法解决问题的时候，还要考虑到执行该算法所需要的存储空间和执行速度等。

一个程序包括对程序中数据和操作的描述：

1) 对数据的描述。在程序中指定所操作数据的数据类型和组织形式，就是数据结构。

2) 对操作的描述。程序执行过程中要执行哪些操作，要经过哪些处理步骤，也就是算法。为此，著名计算机科学家沃斯（Nikiklaus Wirth）对程序用如下公式定义：

程序 = 数据结构 + 算法

在计算机中用于解决一个问题的算法应该具有以下基本特征：

1) 有穷性。一个算法应该包括有限的操作步骤，在有限范围内执行结束，而不能是无限的。

- 2) 确定性。算法的每一个步骤应该是确定的，而不能是模棱两可、含糊不定的。
- 3) 可行性。算法的执行时间和花费的代价是有限的，算法采取的方式是切实可行的。
- 4) 输入。一个算法通常有零个或多个输入，所谓输入就是在算法执行时需要外界提供一个或多个数据。

5) 输出。一个算法可以有一个或多个数据输出。

例如对于下面这个问题，我们可以设计算法如下。

【例 1-1】 求 $1 + 2 + 3 + \cdots + 100$ 的结果。

在这里我们需要用到两个量，①sum 代表累加和，②n 代表累加的数。步骤如下：

- 1) 设置 sum 的初始值为 0，n 的初始值为 1。
- 2) 计算 $\text{sum} + n$ 的结果，用这个结果更新 sum 的值。
- 3) 计算 $n + 1$ 的值，用这个结果更新 n 的值。
- 4) 判断现在的 n，如果 $n \leq 100$ ，跳转到步骤 2)；如果 $n > 100$ ，则算法结束，此时 sum 的值就是所求的结果。

1.1.3 流程图

对于一个程序设计人员，必须能够设计出满足以上特点的算法，并根据算法写出程序。算法通常采用某种语言和规则来描述，常用的有自然语言、流程图和 N-S 图等。

流程图是算法的一种图形化的表示方法，它简洁、直观、清晰的表示算法中每个步骤的操作，有利于人们设计和理解算法，也有利于团队之间的交流与合作。流程图中常用的符号表示形式如表 1-1。

表 1-1 流程图常用符号

图 形	描 述
	程序开始或结束
	计算机步骤/处理符号
	输入/输出指令
	判断和分支
	连接符
	流程线

对于例 1-1 的算法，给出相应的流程图如图 1-1 所示。

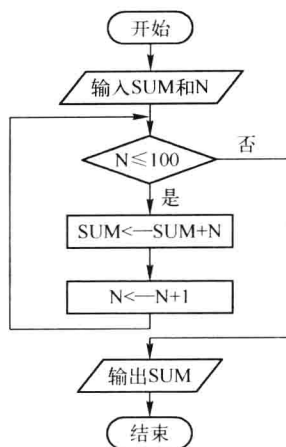


图 1-1 计算 $1 + 2 + 3 + \dots + 100$ 流程图

1.2 C 语言的特点和源程序的基本结构

1.2.1 C 语言的特点

C 语言是当今最流行的几种程序设计语言之一，与其他程序设计语言相比，C 语言具有如下显著的特点：

1) 语言简洁、紧凑，使用方便、灵活。ANSI C 一共只有 32 个关键字，主要用小写字母表示，如表 1-2 所示。

表 1-2 ANSI C 的 32 个关键字

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

2) 运算符丰富。C 语言运算符包含的内容广泛，可以表达复杂的运算规则，并且生成的表达式简练、灵活，有利于提高编译和执行的效率。

3) 数据结构丰富，具有高级语言的各种数据结构。

4) 语法限制不太严格，程序设计的灵活性强、自由度大。

5) 提供了汇编语言的大部分功能，例如可以直接访问内存物理地址，能够进行二进制的运算等。

6) C 语言是结构化的程序设计语言，具有很好的可读性和可移植性等优点。C 语言结构控制语句可以很好地控制程序的结构，使程序结构合理，容易理解，并且 C 语言的语句中没有直接依赖于硬件的语句，因此，用 C 语言编写的程序很容易从一种计算机环境移植到另一种计算机环境。

1.2.2 C 语言程序的构成和书写格式

任何一种程序设计语言，都具有其特定的语法规则和表示形式，程序的构成和书写格式是其表示形式的一个重要方面，按照一定的书写格式和构成规则去书写程序，不仅可以使程序设计人员和读程序的人容易理解，更重要的是，把程序输入到计算机时，计算机能够充分地识别程序，并能够正确地执行它。

下面这个简单的程序展示了 C 语言程序的构成和书写格式。

【例 1-2】 编写程序输出以下信息。

```
This is a C program.
```

程序如下。

```
#include <stdio.h>           //文件包含指令
void main()                  //主函数
{
    printf("This is a C program. \n"); //输出函数调用
}
```

程序的构成和书写格式分析如下：

1) 文件包含指令 `#include <stdio.h>` 是 C 语言中可以使用的预处理指令，不是 C 语言语句，它在程序编译之前进行预先处理操作。该指令的作用是将标准输入输出头文件的内容包含到当前程序当中。

2) 主函数 `main()` 是每个 C 语言程序必须包括的部分，也是 C 语言程序执行的入口，在一个 C 语言程序中除主函数以外还可以定义其他函数，C 语言系统库提供了十分丰富的库函数，无论是 C 的库函数还是用户自定义的函数，我们都可以通过调用函数来实现它的功能，C 语言程序就是由一个一个函数组成的，无论主函数写在程序的什么位置，程序的执行都从主函数开始。在主函数定义中，关键字 `void` 表示函数的类型，即函数返回值的类型，`main()` 的圆括号 `()` 中可以定义参数。`{ }` 中的部分是该函数的函数体。

3) `printf()` 是 C 语言标准库函数中的标准输出函数，该输出语句输出了双引号内部的字符串，因为这里用到了标准输出函数，所以在程序的开头将标准输入输出头文件引入进来。

4) 在程序右侧以双斜线开头的内容是程序的注释部分，对程序进行解释和说明，不参加程序的编译和执行。以双斜线 `//`……开头的内容是单行注释，如果注释的内容需要占用多行，则需要使用 `/* …… */`。

除此之外，在编辑 C 语言程序时还应该注意以下几点：

1) C 语言区分大小写。如 `main()`，不能写成 `Main`、`MAIN` 等。

2) C 语言中的标识符一般采用小写字母，除英文字母外，在标识符中还可以使用数字和下画线，但是数字不能作为开头字符。标识符的长度为 1~32。

3) C 语言程序书写格式灵活，一条语句可以连续写在多行，一行中也可以写多条语句，C 语言是以“`;`”作为语句的结束。

4) 为了使程序的结构清晰、层次分明、可读性好，建议采用“缩进对齐”的格式编辑 C 语言源程序。

1.2.3 C 语言程序的 3 种基本结构

C 语言是结构化的程序设计语言，C 语言程序有如下 3 种基本结构。

1) 顺序结构，如图 1-2 所示，先执行 A 操作，再执行 B 操作，两者是顺序执行的关系。顺序结构是最简单的基本结构。图 1-2 为顺序结构流程图。

2) 选择结构，如图 1-3 所示，先根据菱形框中条件进行判断，条件成立就选择语句或语句块 1 操作执行，不成立则不执行任何操作（如图 1-3a 所示），或执行语句 2 或语句块 2（如图 1-3b 所示）。

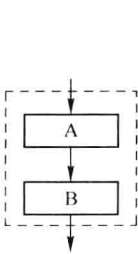


图 1-2 顺序结构流程图

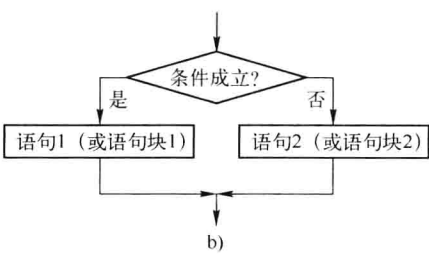
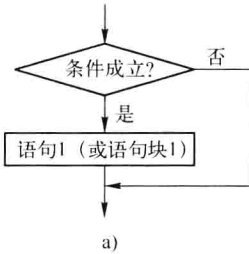


图 1-3 选择结构流程图

3) 循环结构，如图 1-4 所示，循环结构是在某个条件成立的情况下，重复执行同一组操作。循环结构分为当型循环和直到型循环。

当型循环：如图 1-4a 所示，当条件成立时才执行循环体的操作，否则循环体一次也不执行。

直到型循环：如图 1-4b 所示，要先执行一次循环体之后再判断条件表达式是否成立，如果成立，就继续执行循环体，否则就退出循环。

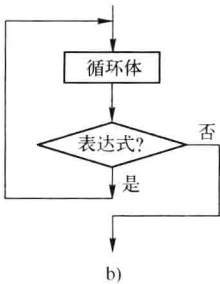
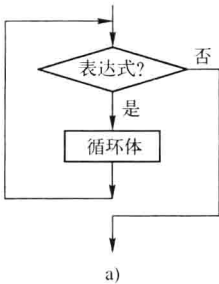


图 1-4 循环结构流程图

a) 当型循环 b) 直到型循环

任何一个 C 语言程序都是属于以上 3 种结构中的一种，或者是由多种结构组合而成，因此，熟练地掌握这 3 种基本结构是学好结构化程序设计语言的关键。

1.3 在 Visual C ++ 6.0 环境开发 C 语言程序

1.3.1 C 语言程序的开发过程

从程序运行方式的角度来划分，C 语言是一种编译型的程序设计语言，开发一个 C 语言

程序要经过编辑、编译、连接和执行 4 个步骤。

(1) 编辑源程序

编辑源程序指的是 C 语言程序的录入和修改。我们可以使用文本编辑器来创建源代码文件，然后以文本文件的形式存放在磁盘上，文件名由用户自行定义，扩展名为 .c，例如 menu.c 等，或者使用文字处理软件来编辑源程序，如写字板、记事本、Word 等，也可以在集成开发环境的编辑器中编辑源程序，如 Borland C++ builder、Visual C++ 等。

(2) 编译预处理

编译预处理是指在对源程序进行编译之前，先对程序中的预处理指令进行处理操作，将预处理得到的信息与程序其他部分一起进行编译，生成目标代码文件。预处理指令不是 C 语言标准的指令，它们以 # 开头，结尾没有分号，以此和 C 语言中标准的指令相区别。现在常用的许多 C 语言开发环境都集成了编译预处理过程，在进行编译时一气呵成。因此不少用户误认为预处理指令是 C 语言标准指令的一部分，甚至以为它们是 C 语句，这是不对的。

(3) 编译源程序

编译是将 C 语言源程序文件翻译成二进制目标程序文件的过程，目标程序文件是用机器语言表示的代码，如果在编译的时候遇到语法错误，则不能生成目标程序，即不能通过编译，需要回到编辑状态修改源程序，直到没有错误为止。

(4) 连接目标程序

编译成功后的目标程序仍然不能执行，需要通过连接程序将编译过的目标程序和程序中用到的函数连接在一起，形成可执行文件，可执行文件的名称与源程序所在的工程名相同，扩展名为 .exe。

(5) 运行可执行文件

经过连接生成的可执行文件即可运行，在 Windows 环境下，可以通过点击可执行文件的名称或图标来执行程序，在 Visual C++ 编译环境中，可以通过运行命令执行程序。C 语言程序的编辑到执行的过程如图 1-5 所示。

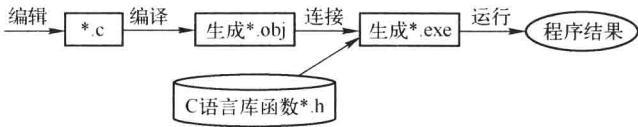


图 1-5 C 语言程序从编辑到执行的过程

1.3.2 Visual C++ 6.0 的启动和操作

目前广泛使用的 C 语言开发环境有 Turbo C，Borland C，Microsoft Visual C++ 6.0 等，全国计算机等级考试（二级 C）采用 Microsoft Visual C++ 6.0 开发环境。为了适应全国计算机等级考试的要求，本书的所有源程序均在 Visual C++ 6.0 环境中编辑、编译和运行。

1. Visual C++ 6.0 的启动

安装好 Visual C++ 6.0 后，通过“开始”→“程序”→“Visual C++ 6.0”→“Visual C++ 6.0”命令启动 Visual C++ 6.0 集成开发环境。如图 1-6 所示。

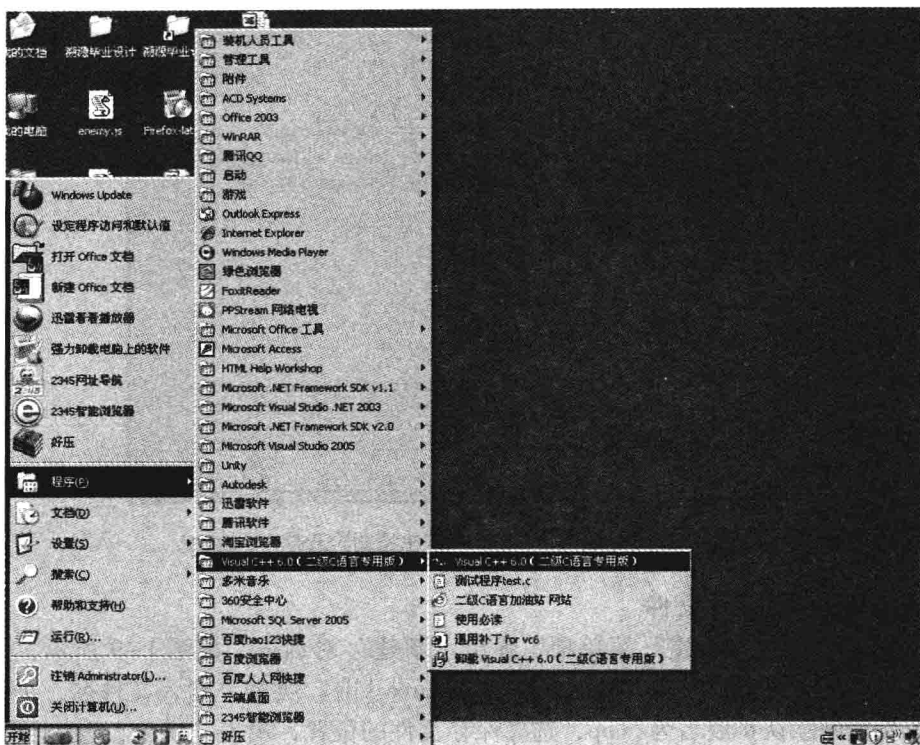


图 1-6 启动 Visual C ++ 6.0

2. 在 Visual C ++6.0 环境中创建工程

进入 Visual C ++6.0 环境以后，通过选择“文件”→“新建”菜单项，弹出“新建”对话框，如图 1-7 所示，在对话框中，可以看到有“文件”、“工程”、“工作区”和“其他文档”4 个选项卡。这里选择“工程”选项卡中的“Win32 Console Application”（控制台应用程序），输入预建立的工程名称，选择工程所在的路径，其他选项按默认值即可，单击“确定”按钮弹出如图 1-8 所示的对话框。在该对话框中选择“一个空工程”选项，在弹出的“新建工程信息”对话框中单击“确定”按钮即可。



图 1-7 Visual C ++ 6.0 “新建”对话框

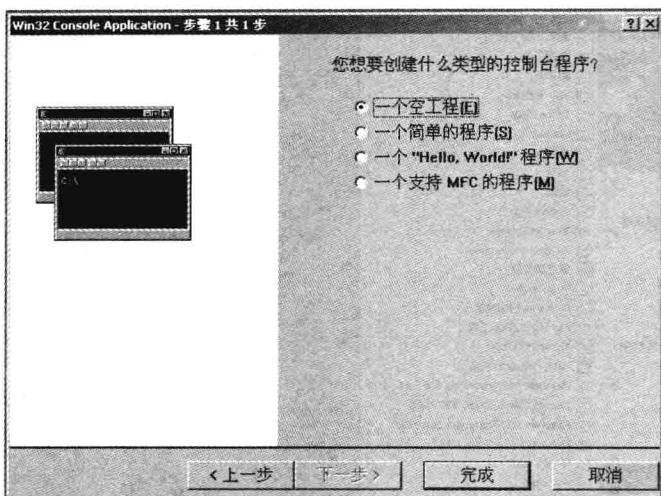


图 1-8 Visual C++ 6.0 “新建控制台程序” 对话框

3. 建立和编辑源程序文件

单击“文件”→“新建”菜单项，弹出“新建”对话框，如图 1-9 所示，选择“文件”选项卡，在文件类型列表中选择“C++ Source File”类型，输入文件名，可加上扩展名.c，若不加则默认扩展名为.cpp，选择保存文件的位置，单击“确定”按钮则生成一个新的空文件 `exe1_1.c`，并弹出源文件编辑窗口，如图 1-10 所示，可以在该窗口中编辑源程序文件。

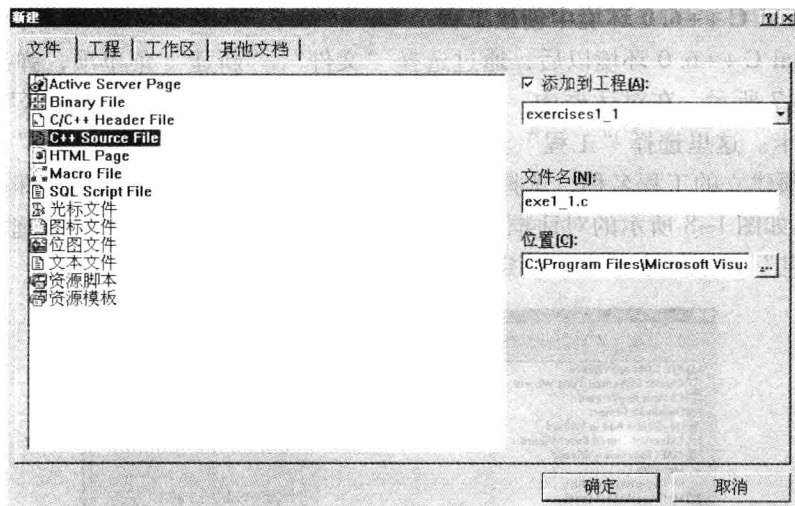


图 1-9 “新建” 对话框

在编辑好源程序后，可以按照 C 语言源程序执行的步骤，首先单击“组建”下拉菜单项，选择“编译”命令，如图 1-11 所示，对文件进行编译，在建立了多个源程序文件以后，要注意编译文件的名称是不是所选文件。编译通过后，继续依次单击“组建”菜单中的“组建”命令和“执行”命令对目标文件进行组建（即连接）和执行，得到结果。