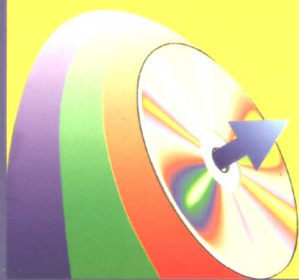


高等院校计算机应用技术系列教材



彭旭东  
王成霞 编著  
万 红

# 程序设计教程(C/C++版)



- ◆ 提供丰富、多样化、实用的教学辅助资料
- ◆ 赠送教师完整的电子教案及配套的教学辅导书



清华大学出版社

高等院校计算机应用技术系列教材

# 程序设计教程 (C/C++版)

彭旭东 王成霞 万红 编著

清华大学出版社

北 京

## 内 容 简 介

在这个科学技术飞速发展的时代,程序设计是大学生在使用计算机时必须具备的能力。本书面向广大非计算机专业的程序设计初学者,由浅入深地讲述了C语言的设计方法。同时,还阐述了程序设计的基本方法和面向对象程序设计的基本知识。

本书根据初学者的特点,按照学生的认知规律精心策划,力求使用通俗易懂的语言、丰富的图形及例题介绍比较抽象的概念和逻辑。对于其中比较难于理解的算法,还提供了动画演示(可通过<http://www.tupwk.com.cn/downpage/index.asp>下载)。

本书内容深入浅出,循序渐进,便于自学。可作为非计算机专业C程序设计课程的教材,也可以作为相关培训班的教材。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

### 图书在版编目(CIP)数据

程序设计教程(C/C++版)/彭旭东,王成霞,万红编著. —北京:清华大学出版社,2005.6

(高等院校计算机应用技术系列教材)

ISBN 7-302-10748-3

I. 程… II. ①彭…②王…③万… III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2005)第028278号

出版者:清华大学出版社 地 址:北京清华大学学研大厦  
http://www.tup.com.cn 邮 编:100084  
社总机:010-62770175 客户服务:010-62776969

组稿编辑:胡伟卷

文稿编辑:刘金喜

封面设计:王永

版式设计:康博

印刷者:北京市世界知识印刷厂

装订者:三河市化甲屯小学装订二厂

发行者:新华书店总店北京发行所

开本:185×260 印张:29 字数:670千字

版次:2005年6月第1版 2005年6月第1次印刷

书号:ISBN 7-302-10748-3/TP·7159

印数:1~5000

定 价:39.00元

# 前 言

“程序设计”是针对广大非计算机专业的大学本科学生开设的一门必修课，也是计算机学科的基础课程。本书是为“程序设计”课程编写的教材，其内容选取符合教学大纲的要求，同时也兼顾了学科的广度和深度，适用面广。

本书共分3部分19章。第1部分是入门篇，包含前9章。第1~2章从介绍程序设计开始，首先为学习本课程提供了指导性的建议，然后讨论了结构化程序设计的方法和程序调试方法。第3章介绍了程序的各个组成部分以及程序的运行过程。第4章围绕简单的数据计算介绍了数据类型、常量、变量、运算符和表达式等基本概念，以及数据输入输出的方法。第5~7章介绍了程序控制结构的概念、选择结构和循环结构。作为入门篇的高级部分，第8章和第9章讨论了数组和字符串的概念以及应用。

第2部分是进阶篇，包含第10~16章。其中，第10~15章依次讨论了C语言中的指针、函数、位运算、复合数据类型、文件和编译预处理等6方面的内容。第16章围绕着几个实例，介绍了C++与面向对象程序设计中的基本概念。

第3部分是应用篇，用3章的篇幅详细讨论了链表与堆栈两种数据结构，以及排序与查找两类基本算法。

在组织内容时，我们围绕两条主线编写，即语法规则和程序设计。在编写的过程中，一方面吸收了国内外多本经典教材的特点，另一方面也融会了小组成员多年丰富的教学经验和工程开发经验。本教材适合非计算机专业的本科生阅读，同时也可供广大的C语言程序设计初学者和爱好者阅读。

本教材提供了示例程序源代码、重点算法的Flash动画、交互式的自测题、供教师授课时使用的幻灯片、供学生课下自学和复习使用的Web站点以及其他素材。读者可通过<http://www.tupwk.com.cn/downpage/index.asp>下载这些内容。

作为补充资料，本教程还有配套的《上机指导和习题解析》。其中包括本教程的习题解答、常见编译系统的上机指导、上机实验和指导、补充习题和模拟试题。如果能够与本教程同时使用，将起到更好的教学效果。

本书的第10~13章由王成霞老师编写；第4、14、15、19章、附录A和B、第17章中的17.2节以及该章与堆栈有关的建议和练习题由万红老师编写；其余部分由彭旭东老师编写。

在编写本书的过程中，我们得到了天津理工大学计算机科学与工程系领导的大力支持，清华大学出版社的胡伟卷女士、天津理工大学的莫秀良老师也为本书的完成提供了大量的帮助，在此一并表示感谢。

作 者

# 目 录

## 第 I 部分 入门篇

<b>第 1 章 概述</b> .....1	3.2.3 函数.....22
1.1 程序设计的分类.....1	3.2.4 注释.....23
1.1.1 程序设计语言的分类.....1	3.3 从编辑到运行.....24
1.1.2 面向过程和面向对象.....2	3.4 建议.....26
1.2 C 语言.....3	3.5 练习题.....27
1.2.1 发展历史.....3	<b>第 4 章 简单的数据计算</b> ..... 28
1.2.2 特点.....4	4.1 C 的数据类型.....28
1.3 学习方法建议.....4	4.1.1 数据类型.....28
1.3.1 学习计划.....5	4.1.2 基本数据类型的存储方式 和取值范围.....29
1.3.2 参考资料的介绍.....5	4.2 常量与变量.....31
1.4 常见的 C/C++ 编译系统.....6	4.2.1 常量.....31
1.5 建议.....7	4.2.2 变量.....34
1.6 练习题.....7	4.3 算术运算符.....37
<b>第 2 章 结构化程序设计与小规模   软件设计初步</b> .....8	4.3.1 二元算术运算符.....37
2.1 模块及其特点.....8	4.3.2 一元算术运算符.....38
2.2 软件工程和软件生命期.....9	4.4 赋值运算符.....39
2.3 软件规模引起的问题.....10	4.5 表达式.....39
2.4 小规模软件的设计.....11	4.5.1 运算符的优先级和结合性.....40
2.5 示例.....14	4.5.2 算术表达式.....40
2.6 建议.....17	4.5.3 赋值表达式.....42
2.7 练习题.....17	4.6 数据的输入与输出.....43
<b>第 3 章 C 程序设计基础</b> .....18	4.6.1 数据的输出.....44
3.1 再看“您好, 张先生”.....18	4.6.2 数据的输入.....47
3.2 程序组成部分.....20	4.7 示例: 计算温度.....50
3.2.1 基本单词.....20	4.8 建议.....51
3.2.2 语句.....22	4.9 练习题.....52
	4.10 难点解析: 再谈数据类型 问题.....54
	4.10.1 数据类型.....54

4.10.2	赋值时的数据类型	57	7.11	练习题	108
4.10.3	整型数据的输出格式	60	<b>第 8 章</b>	<b>数组及其应用</b>	<b>113</b>
4.10.4	强制类型转换	60	8.1	概述	113
4.10.5	sizeof 运算符	61	8.2	数组的定义	114
<b>第 5 章</b>	<b>程序控制结构基础</b>	<b>62</b>	8.3	访问数组	115
5.1	概述	62	8.4	初始化数组	116
5.2	复合语句和空语句	64	8.5	二维数组	118
5.3	关系运算符和关系表达式	65	8.5.1	定义	119
5.4	逻辑运算符和逻辑表达式	66	8.5.2	访问	119
5.5	示例	67	8.5.3	初始化	121
5.6	建议	68	8.6	示例	125
5.7	练习题	68	8.7	建议	130
<b>第 6 章</b>	<b>选择结构</b>	<b>69</b>	8.8	练习题	131
6.1	概述	69	<b>第 9 章</b>	<b>字符与字符串</b>	<b>135</b>
6.2	if-else 语句	69	9.1	字符	135
6.3	嵌套的 if-else 语句	73	9.2	字符的输入与输出	137
6.4	switch-case 语句	77	9.2.1	使用 printf 和 scanf	137
6.5	break 语句	80	9.2.2	使用其他输入输出函数	139
6.6	? :运算符	82	9.3	字符与整数	142
6.7	示例	82	9.4	字符串	143
6.8	建议	86	9.5	数组与字符串	145
6.9	练习题	86	9.6	字符串的输入与输出	147
<b>第 7 章</b>	<b>循环结构</b>	<b>89</b>	9.7	处理字符串	151
7.1	概述	89	9.8	示例	156
7.2	for 语句	90	9.9	建议	158
7.2.1	逗号运算符和逗号表达式	90	9.10	练习题	158
7.2.2	for 语句	91	<b>第 II 部分</b>	<b>进 阶 篇</b>	
7.3	while 语句	96	<b>第 10 章</b>	<b>指针</b>	<b>161</b>
7.4	do-while 语句	98	10.1	地址与指针	161
7.5	嵌套的循环语句	100	10.1.1	地址与指针的概念	162
7.6	3 个特殊语句	102	10.1.2	指针变量	163
7.7	死循环	104	10.2	指针的运算	170
7.8	编写循环结构容易犯的错误	105	10.2.1	指针的移动	170
7.9	示例	106			
7.10	建议	108			

10.2.2	两个同类型指针相减	173	11.6.1	变量的作用域	241
10.2.3	指针的比较	173	11.6.2	变量的存储类型	244
10.3	指针与一维数组	174	11.7	嵌套调用和递归调用	251
10.3.1	一维数组与指针的关系	174	11.7.1	嵌套调用	251
10.3.2	用指针访问一维数组 的元素	176	11.7.2	递归调用	253
10.4	指针与二维数组	179	11.8	函数指针	256
10.4.1	二维数组元素的地址	179	11.9	main 函数	259
10.4.2	用指针变量访问二维数组 元素	182	11.10	函数的存储类型	262
10.4.3	用行指针变量访问二维 数组元素	184	11.10.1	内部函数	262
10.5	指针与字符串	186	11.10.2	外部函数	262
10.5.1	用字符数组处理字符串	186	11.11	示例	265
10.5.2	用字符指针处理字符串	187	11.12	建议	268
10.5.3	字符指针和字符数组 的比较	189	11.13	练习题	269
10.5.4	常用的字符串处理函数	192	<b>第 12 章 位运算</b>		<b>276</b>
10.6	二级指针与指针数组	196	12.1	概述	276
10.6.1	二级指针	196	12.2	位运算符	276
10.6.2	指针数组	197	12.2.1	按位逻辑运算符	277
10.7	内存空间的动态分配	202	12.2.2	移位运算符	279
10.8	示例	206	12.3	示例	281
10.9	建议	207	12.4	建议	284
10.10	练习题	210	12.5	练习题	284
<b>第 11 章 函数</b>		<b>216</b>	<b>第 13 章 复合数据类型</b>		<b>287</b>
11.1	概述	216	13.1	结构类型	287
11.2	函数的定义与说明	219	13.1.1	结构类型和结构变量 的定义	288
11.2.1	函数的定义	219	13.1.2	结构变量的引用和结构 变量的初始化	291
11.2.2	函数的说明	222	13.1.3	结构数组	295
11.3	函数的调用过程	224	13.1.4	结构指针	298
11.4	函数的参数	227	13.1.5	结构变量和结构指针作 函数参数	302
11.4.1	在函数间传递变量	227	13.1.6	返回结构和结构指针 的函数	308
11.4.2	在函数间传递数组	231	13.2	共用体	311
11.5	函数的返回值	240	13.2.1	共用体的定义与引用	311
11.6	变量的作用域和存储类型	241			

13.2.2	共用体的引用	312	15.2	宏定义	368
13.2.3	共用体与结构的比较	313	15.2.1	不含参数的宏定义格式	368
13.2.4	共用体应用举例	314	15.2.2	含有参数的宏定义格式	369
13.3	位段结构	315	15.2.3	宏的作用域和宏的重新定义	372
13.4	枚举	319	15.2.4	宏定义的嵌套	373
13.5	类型定义	322	15.2.5	宏扩展中的优先级问题	373
13.6	示例	325	15.3	文件包含	375
13.7	建议	328	15.3.1	文件包含的格式	375
13.8	练习题	328	15.3.2	文件包含的两种方式	376
<b>第 14 章</b>	<b>文件</b>	<b>334</b>	15.3.3	文件包含的嵌套	378
14.1	概述	334	15.4	条件编译	378
14.1.1	文件的定义	334	15.5	示例	382
14.1.2	文件的逻辑结构	335	15.6	建议	383
14.1.3	两种文件系统	336	15.7	练习题	383
14.1.4	常用缓冲文件系统函数	337	<b>第 16 章</b>	<b>C++与面向对象程序设计</b>	<b>386</b>
14.2	打开与关闭文件	337	16.1	概述	386
14.2.1	文件型指针	337	16.2	类和对象	388
14.2.2	文件的打开	338	16.2.1	类	388
14.2.3	文件的关闭	340	16.2.2	对象	392
14.3	读写文件	341	16.3	消息和方法	394
14.3.1	字符读写函数	342	16.4	继承	402
14.3.2	字符串读写函数	343	16.5	多态	410
14.3.3	格式化读写函数	346	16.6	练习题	411
14.3.4	数据块读写函数	347	<b>第III部分 应用篇</b>		
14.4	文件的定位	351	<b>第 17 章</b>	<b>链表和堆栈</b>	<b>413</b>
14.4.1	rewind 函数	351	17.1	链表	413
14.4.2	ftell 函数	353	17.1.1	链表概述	413
14.4.3	fseek 函数和随机读写	353	17.1.2	问题	415
14.5	检测文件	357	17.1.3	创建链表	416
14.6	非缓冲文件的操作	358	17.1.4	显示链表	419
14.7	示例	361	17.1.5	插入结点	419
14.8	建议	362	17.1.6	删除结点	422
14.9	练习题	362	17.2	堆栈概述	424
<b>第 15 章</b>	<b>编译预处理</b>	<b>367</b>			
15.1	概述	367			

---

17.3	建议	426
17.4	练习题	426
<b>第 18 章</b>	<b>排序</b>	<b>428</b>
18.1	概述	428
18.2	插入排序	429
18.3	选择排序	431
18.4	冒泡排序	433
18.5	归并排序	435
18.6	建议	438
18.7	练习题	438
<b>第 19 章</b>	<b>查找</b>	<b>439</b>
19.1	概述	439
19.2	顺序查找	439
19.3	折半查找	442
19.4	建议	445
19.5	练习题	446
<b>附录 A</b>	<b>ASCII 码</b>	<b>448</b>
<b>附录 B</b>	<b>运算符的优先级和结合性</b>	<b>449</b>
<b>参考文献</b>		<b>450</b>

# 第 I 部分 入 门 篇

## 第1章 概 述

本章是对程序设计课程的概述，其中不但介绍了程序设计和 C 语言，还为广大的程序设计初学者提供了指导性的学习建议。通过学习本章内容，读者应在掌握了有关程序设计和 C 语言的基本常识之后，能够制定出适合自己的学习方法。

**本章的学习目标：**

- 程序设计的分类
- 选用 C 语言进行程序设计的理由
- 学习程序设计的方法

### 1.1 程序设计的分类

程序设计是计算机应用中很大的一个分支，本节只讲述其中两个比较基本的问题：程序设计语言和程序设计方法。

#### 1.1.1 程序设计语言的分类

程序设计语言是人们控制计算机的工具，程序员可以利用它编写出各种各样的程序。一般把用程序设计语言编写的程序称为源程序。根据语言的功能和特点，可以将程序设计语言分成 4 类：高级语言、初级语言、低级语言和机器语言。目前流行的 Visual Basic、Java、Delphi 等语言都属于高级语言。对于 C 语言，有人习惯上也将其划分成高级语言，但是由于它具有许多低级语言的功能，所以本书把它划分到初级语言中。汇编语言属于低级语言。程序员、程序设计语言和计算机之间的关系如图 1-1 所示。

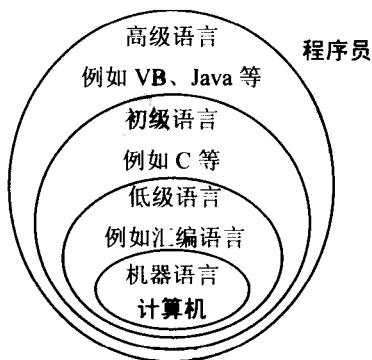


图 1-1 程序员、程序设计语言和计算机之间的关系

机器语言由机器指令组成，使用二进制形式编写，能够直接被 CPU 理解和执行。但由于每一种型号的计算机都有自己特定的指令系统，所以，为一种计算机编写的程序很难在另外一种计算机上运行。我们将这样的程序评价为可移植性非常差。此外，机器语言程序的可读性、可修改性较差。

相比之下，建立在机器语言之上的汇编语言就方便多了。它把机器语言中的指令都转换成了接近自然语言的助记符形式，例如使用“ADD”表示加法，使用“SUB”表示减法等。虽然人们容易识别和记忆符号，但是计算机只能识别 0 和 1。因此，用汇编语言编写的源程序必须使用汇编程序将其翻译成计算机能够识别的目标程序才可以执行。由于汇编语言也是一种面向机器的语言，它与机器指令基本上是一一对一的关系，因此它的可移植性也非常差。

以 Basic、FORTRAN 等语言为代表的高级语言充分吸收了自然语言和数学语言的特点，以面向程序员的语句取代了面向计算机的指令。这样不但提高了人们学习程序设计语言的速度，大幅度提高了编写、维护程序的效率，比较好地解决了软件可移植性差的问题，同时也提高了程序的质量。从某种意义上讲，高级语言的发展也为计算机在各个领域的普及，以及解决更大规模的问题奠定了基础。然而，也正因为高级语言不是面向计算机的，所以在执行时必须借助于编译程序或者解释程序将其转换成计算机能够识别的二进制形式才能执行。由于高级语言的语句和机器语言的指令是一对多的关系，所以执行的速度相对较低一些。

以 C 语言为代表的初级语言不但拥有高级语言的简单易学、可移植性强的优点，同时也具备直接访问硬件的能力，其程序的执行速度接近于用汇编语言编写的程序。这是许多高级语言所不具备的优点，也正因为如此，C 语言一经推出就得到了广泛的好评。

### 1.1.2 面向过程和面向对象

目前流行的程序设计方式主要是面向过程程序设计和面向对象的程序设计。这是两种截然不同的设计方式。面向过程的语言主要有 C 和 FORTRAN 等。使用这种语言编写程

序需要考虑的是问题的求解方法。它以计算机为核心,要求程序员按照计算机处理问题的方式来描述、表达问题和解决过程。使用面向过程的程序设计方式时,研究的重点是数据结构和算法,也就是如何精确地描述问题,如何准确、高效率地解决问题。因此,面向过程的程序设计比较适合于小规模的程序,适合用来解决相对简单的问题。

如果程序的规模非常大,或者要解决的问题非常复杂,可以使用面向对象的程序设计方式。除了具体的解题过程以外,面向对象程序设计更强调从整体考虑问题的各个组成部分、它们的各种状态以及状态的切换。常见的面向对象的程序设计语言有 C++ 和 Java 等。第 15 章以 C++ 语言为例对此进行比较详细的讨论。

## 1.2 C 语言

C 语言是国际上广泛流行的计算机程序设计语言。它是一种面向过程的结构化程序设计语言,也是一种初级语言。它不但适合编写系统软件,也同样适合编写各类应用软件。下面从其发展历史和特点两个方面加以介绍。

### 1.2.1 发展历史

C 语言是为了开发 UNIX 操作系统而产生的。当时的程序设计语言或者是以汇编语言为代表的面向机器的语言,或者是以 ALGOL60 为代表的高级语言。这两种语言都不能同时满足直接访问硬件和可移植性好的要求,因此不适合用来开发操作系统。后来出现的 CPL、BCPL 和 B 等语言虽然都能够兼顾这两个要求,但它们有的过于庞大而难于实现,有的过于简单而功能有限,所以都没有得到广泛使用。

1972 年到 1973 年,美国贝尔实验室的 D.M.Ritchie 在 BCPL 和 B 语言的基础之上开发了 C 语言,并于 1973 年和贝尔实验室的 K.Thompson 合作,把 UNIX 中 90% 以上的代码用 C 语言改写,取得了成功。以后,随着 C 语言的改进,UNIX 操作系统迅速在各种机器上得到了实现,并得到了广泛使用。UNIX 的成功也引起了人们对 C 语言的注意。其简洁的格式、强大的功能和良好的可移植性,使 C 语言受到了人们的普遍赞誉,世界各地的程序员都使用它编写各种程序。

为了统一各组织所用 C 语言版本间的差异,美国国家标准化组织(ANSI)于 1983 年成立了一个委员会,对 C 语言进行了标准定义,这就是 ANSI 标准 C 语言。

现在 C 编译系统中的绝大多数都支持 ANSI 标准 C 语言,但它们也都进行了不同程度的扩充。本书中所有的 C 语言示例程序都在 Turbo C 2.0 中调试通过,所有的 C++ 示例程序都在 Visual C++ 6.0 中调试通过。如果要使用其他编译系统运行这些程序,请参考有关手册。

## 1.2.2 特点

目前流行的程序设计语言有许多种,例如 C、Perl、Basic、FORTRAN、Java、Delphi、C#等。这些都是非常优秀的程序设计语言。除了可以用来完成普通的程序设计任务以外,它们也都有自己独特的优势。尽管如此,本教程仍然选择 C 语言讲述程序设计,因为 C 语言具有以下特点:

- 功能强大。由于 C 语言是初级语言,它既包含有高级语言的特点,又能够直接访问硬件,因此,从字处理、某种语言的编译系统、图形图像处理、网络通信,一直到操作系统,从相对简单的单机系统到比较复杂的分布式系统和执行并行计算的系统,各类项目都可以使用 C 语言很好地完成。因此,国外曾经有人说“使用 C 语言能够完成的工作只受限于您的想象力”,国内也有“真正的程序员使用 C”的说法。
- 语言简洁、灵活。从表面看,C 语言使用的单词、术语很少,ANSI 标准 C 中只规定了 32 个。我们将这些词称为保留字,C 语言在保留字的基础之上构造了各项功能。对于运算符,C 语言也进行了简化。例如对于程序中大量出现的  $i=i+1$  简化成了  $i++$ ,类似  $a=a+b$  的语句简化成了  $a+=b$ 。从更高的层次看,C 语言将许多功能都用函数实现,即便是最常用的输入和输出功能也是如此。这样就能够根据需要配置和扩充库函数,从而不但保证了可移植性,而且还保证了编译后的可执行文件具有更小的体积。
- 可移植性强。C 语言得到迅速普及的原因之一就是可移植性强。从单板机到 IBM PC 兼容机,从图形工作站到小型机、大型机,从 Microsoft Windows 操作系统到 Linux 和 UNIX 操作系统,都可以看到 C 语言的影子。而且在一个系统上编写的程序不加以修改或者只需要进行少量的修改就可以在另外一个系统上运行。
- 面向过程的结构化程序设计语言。前面曾经比较过面向过程和面向对象的程序设计,这里就不再详细介绍了。

## 1.3 学习方法建议

程序设计是一门实践性强、趣味性强的课程,如果没有比较好的学习方法往往会事与愿违。因此,本节围绕教程的内容为广大的初学者提供学习建议。由于读者之间存在各种各样的差异和要求,所以这里的建议仅供参考,不强迫每个人都严格遵守。

在学习时,如果读者有某种程序设计的基础,应该仔细体会它与 C 语言的细微差别,体会例题、练习题和建议中体现出来的程序设计思想以及注意事项。另外,在学习时,应该考虑使用尽可能多的方法完成例题、练习题。

如果读者没有基础,应该先以每章示例前面的内容为主,掌握了这些基本概念之后,学习示例,然后完成练习题。在系统学习每章的基本内容之后,再阅读每章后面的“建议”,

并尝试修改以前的作业。

本教程有配套的《程序设计教程(C/C++版)上机指导和习题解析》(后简称为《上机指导和习题解析》)和其他配套的教学资料。如果能够配合教材同时使用,将起到更好的教学效果。

### 1.3.1 学习计划

根据学生的具体情况和学习要求,建议本课程使用 60 到 80 学时,其中上机实验至少需要 20 学时。为了达到更好的教学效果,建议课上、课下学习时间之比为 1: 1 到 1: 2。

对于第 4 章后面的“难点解析”部分,建议在完成入门篇之后学习。其余内容,可以按照章节顺序依次学习。为了保证教学效果,各章的学时安排如下:

授课学时:第 4、10、11、13、16 章使用 4 学时完成,其余每章可以使用大约 2 个学时完成。

上机学时:第 3、4、6~11、18 和 19 章每章至少安排 2 学时上机,建议第 13、14 章每章安排 2 学时,第 16、18 章每章安排 4 学时的上机实验。

其中第 2 章和第 16 章的内容可以选学。

### 1.3.2 参考资料的介绍

虽然我们试图把 C/C++ 语言中的重点概念和程序设计的初步概念讲述清楚,但是毕竟这是适合于初学者的入门教程,因此部分内容讲述得不够透彻。接下来介绍几本国内外著名的优秀教材,供读者学习时补充参考。

Brian W. Kernighan 和 Dennis M. Ritchie 著的《The C Programming Language(Second Edition)》以浅显的语言对 C 语言进行了全面、准确、透彻的描述。虽然篇幅很短(正文部分不足 190 页,而且是 32 开本),但是在普通教材中叙述得不够清楚的问题都可以在这里找到很明确的答案。这不愧是世界公认的 C 语言经典著作。想更系统地学习 C 语言的人可以参考这本书。

本教程中 C++ 部分的内容不多,如果对这种语言感兴趣,想作进一步了解,有很多教材可供参考。但如果想深入、系统地学习这种语言,可以参考 Bjarne Stroustrup 著的《The C++ Programming Language(Special Edition)》。和前一本教材类似,这也是世界公认的权威著作。教材不但详细介绍了 C++ 本身,而且还融合了作者多年丰富的软件开发经验。因此阅读本书,有一种和大师进行面对面交流的感觉。初学者能够从开始得到正规、系统的训练。有一定基础的人阅读本书也能够从中获益匪浅。

本书中的“应用篇”涉及了数据结构的部分内容。如果对这部分内容感兴趣,或者希望进一步提高程序设计能力,可以参考严蔚敏、吴伟民著的《数据结构(C 语言版)》。作为计算机专业的教材,该书以浅显易懂的语言和丰富的图例、动画演示详细介绍了常用的数据结构和算法。该书自从 1997 年出版发行以来,至今已累计发行了 100 万册,也是一本不

可多得的好书。

如果想在更高的层次上研究有关程序设计的问题，可以阅读有关软件工程的书籍。

此外，和本教程配套的《上机指导和习题解析》也是非常好的参考书。它不但提供了教材中的习题答案、常见编译系统的上机指导，还针对初学者安排了上机实验。为了巩固学习效果，我们还特意增加了大量的补充习题和模拟试题。如果能够与本教材配套使用，将会起到很好的教学效果。

## 1.4 常见的 C/C++ 编译系统

C/C++ 是国际上非常流行的程序设计语言，它的流行也体现在拥有众多的编译系统。仅在 DOS 和 Windows 操作系统中，比较著名的就有 Turbo C、Microsoft C、Borland C++、Visual C++ 以及 C++ Builder 等。教学中通常使用的是 Turbo C 2.0 和 Visual C++ 6.0。

Turbo C 2.0 是 Borland 公司开发的、在 DOS 操作系统中运行的一个优秀的 C 语言系统。在 Windows 操作系统中，它既能够以全屏幕方式运行，也能够以窗口方式(如图 1-2 所示)运行。

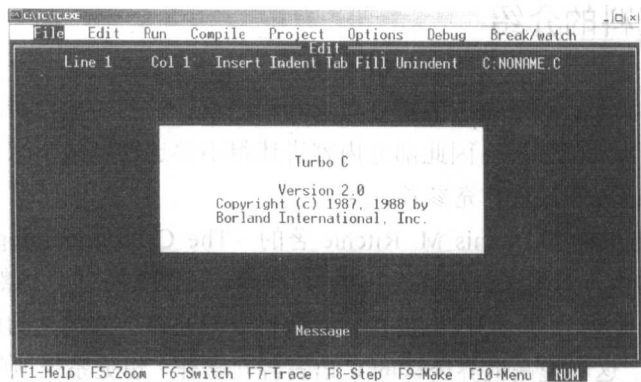


图 1-2 以窗口方式运行的 Turbo C 2.0

本书介绍 Turbo C 的第一个原因在于它是一个高度集成的开发环境。在这个环境中，可以很方便地进行程序的编辑、编译、连接、调试和运行。此外，它在支持 ANSI C 标准的基础之上还有所扩充，提供的库函数既包括 ANSI C 的标准函数库，还包括一批扩充库函数。

介绍 Turbo C 的另一个原因是它更适合 C 程序设计的课程教学。整个系统比较简单，但是功能却非常完整，需要初学者掌握的概念比较少，容易入门。Turbo C 系统对计算机要求非常低，即使在 IBM PC 机(640KB 内存，双软驱，无硬盘)上也可以很好地运行，而且编译速度快，很适合做练习和一般的程序开发。

Visual C++ 6.0 是 Microsoft 公司于 90 年代末推出的一个基于 C/C++ 的集成开发工具，它运行在 Windows 操作系统下，是 Windows 操作系统中非常优秀的开发工具。从实用的角度讲，推荐初学者使用 Visual C++ 学习程序设计。

有关上述两个开发工具的上机步骤和调试方法，请参见教程配套的《上机指导和习题解析》。

## 1.5 建 议

程序设计是门实践性非常强的课程，因此必须保证有足够的时间上机调试程序。只有这样才能更快更好地掌握相关概念，并培养自己的学习兴趣。

## 1.6 练 习 题

### 一、选择题

1. 程序设计语言分成\_\_\_\_\_。  
A) 初级语言、中级语言和高级语言      B) 机器语言和人类语言  
C) 机器语言、低级语言和高级语言      D) 低级语言和高级语言
2. 下面对 C 语言描述错误的是\_\_\_\_\_。  
A) 语法简单但是功能强大      B) 可以在很多软、硬件系统中使用  
C) 它是一种面向过程的程序设计语言      D) 它是在 C++的基础上简化得到的

### 二、上机题

1. 在自己的计算机上安装和设置 Turbo C 2.0 和 Visual C++6.0。
2. 输入并运行下面的程序，熟悉开发环境。

```
#include <stdio.h>
main()
{
    printf("This is an example.\n");
}
```

# 第2章 结构化程序设计与小规模 软件设计初步

目前，计算机已经普及到了社会的各个领域。为了顺应工作和科研的需要，越来越多的非计算机专业的科研工作者需要编写一些实用性非常强的程序软件。而开发软件或者说程序设计绝不等同于学习一门程序设计语言。它需要一个科学的模式、一种严谨的过程来保证软件的质量和开发的效率。

鉴于此，我们编写了本章内容，目的是拓宽初学程序设计的人的思路，使其能够站在更高的层次看待程序设计，在面对规模比较大的程序时，能够以科学的方法进行设计。对软件的质量和开发效率有更高要求的读者，可以参考有关软件工程的教程。

**本章的学习目标：**

- 自顶向下的软件开发过程
- 软件生命周期
- 小规模软件的设计方法
- 模块的概念和特点

## 2.1 模块及其特点

按照规模，可以将软件粗略地划分成3大种：微型软件、小型软件和中型及以上规模的软件。所谓微型软件指的是1个人在半小时到1个月之内能够完成的软件，软件源程序通常不超过500行语句。这类软件通常是实验数据的简单处理等。如果1个人需要半年以内的时间完成，而且源程序的规模不超过2000行，那么这种软件应该属于小型软件。工作和科研项目中规模略大一些的数值计算、数据处理等问题属于此类问题。

如果软件的规模继续扩大，这时的软件就属于中型及以上规模的软件了。这种规模的软件由专业程序员开发，他们所用的程序设计方法超出了本教程的范围，本书对此不与讨论。另外，为了叙述方便，本书将微型软件和小型软件都统称为小规模软件，将中型及以上规模的软件统称为大规模软件。

对于小规模软件来讲，除非待处理的问题非常复杂，必须要使用面向对象的程序设计方法解决，否则使用面向过程的方法就可以很好地解决问题。