

SAMS



附光盘

21天学通 C语言 (第6版)

[美] Bradley L. Jones 著
Peter Aitken 译
信达工作室

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

21 天学通 C 语言 (第 6 版)

[美] Bradley L. Jones Peter Aitken 著

信达工作室 译

人民邮电出版社

图书在版编目 (CIP) 数据

21 天学通 C 语言 / (美) 琼斯 (Jones, B. L.), (美) 艾特肯 (Aitken, P.) 著; 信达工作室译.
—北京: 人民邮电出版社, 2003.3

书名原文: Teach Yourself C in 21 Days

ISBN 7-115-11144-8

I. 2... II. ①琼... ②艾... ③信... III. C 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 008590 号

版权声明

Bradley L. Jones, Peter Aitken: Sams Teach Yourself C in 21 Days, Six Edition (ISBN:0672324482)

Copyright © 2003 by Sams Publishing.

Authorized translation from the English language edition published by Sams.

All rights reserved.

本书中文简体字版由美国 Sams 授权人民邮电出版社出版, 未经出版者书面许可, 对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

21 天学通 C 语言 (第 6 版)

-
- ◆ 著 [美] Bradley L. Jones Peter Aitken
译 信达工作室
责任编辑 李 际

 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67132705
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

 - ◆ 开本: 787×1092 1/16
印张: 32
字数: 1 041 千字 2003 年 3 月第 1 版
印数: 1-5 000 册 2003 年 3 月北京第 1 次印刷

著作权合同登记 图字: 01-2002-5929 号

ISBN 7-115-11144-8/TP · 3357

定价: 52.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

内容提要

本书译自《Teach Yourself C in 21 Days》第6版，该书的前五版都登上了畅销书排行榜，是初学者学习 C 语言的经典之作。本版按最新的标准（ISO/IEC:9899-1999），以循序渐进的方式介绍了 C 语言编程方面的知识，并提供了丰富的实例和大量的练习。通过学习实例，并将所学知识用于完成练习，读者将逐步了解、熟悉并精通 C 语言。

本书包括四周的课程，前三周详细介绍了 C 语言，第四周是附加课程，简要地介绍了最流行的面向对象语言——C++、Java 和 C#，附加课程的内容在光盘中以 pdf 格式文件提供。第一周的课程介绍了 C 语言程序的基本元素，包括变量、常量、语句、表达式、函数和循环；第二周介绍了数组、指针、字符和字符串、结构和共用体、变量的作用域、输入/输出等；第三周介绍了有关指针和函数的高级主题、磁盘文件读写、字符串操纵函数、函数库、内存管理以及编译器的高级用法等。

本书是为初中级程序员编写的，可作为学习 C 语言的教程或参考资料。

作者简介

Bradley L. Jones 在 internet.com 网站就职，负责管理 EarthWeb 软件开发频道，包括诸如 Developer.com、CodeGuru.com 和 Gamelan.com 等网站。Bradley 领导开发了用于各种平台（从 Palm OS 到大型机系统）的小型和分布式系统。Bradley 拥有使用 C、C#、C++、XML、SQL Server、PowerBuilder、Visual Basic、ASP 和 Satellite Forms 等工具进行开发工作的经验。他编写的其他图书包括《Teach Yourself Advanced C in 21 Days》和《21 天学通 C#》。

Peter Aitken 拥有十多年计算机和编程方面的写作经验，编写的图书大约有 30 本，并在计算机杂志上发表过几百篇文章。他最近编写的图书有《Visual Basic .NET Programming With Peter Aitken》、《Office XP Development With VBA》、《XML the Microsoft Way》、《Windows Script Host》和《Teach Yourself Visual Basic .NET Internet Programming in 21 Days》等。Peter Aitken 曾担任《Visual Developer》杂志的撰稿编辑多年，负责编写 Visual Basic 专栏；另外，他还经常在《Microsoft OfficePro》杂志和网站 DevX 上发表文章。从 1994 年起，Peter Aitken 一直负责经营 PGA 咨询公司，为企业、学术和政府机构开发应用程序和 Internet 程序。

本书旨在引导读者在 21 天内学通 C 语言编程。虽然有来自诸如 C++、Java 和 C# 的激烈竞争，但很多初学编程者还是会选择 C 语言。基于第 1 天课程介绍的原因，选择 C 语言可确保您不会误入“歧途”。

将本书作为自学 C 语言的教材是一个明智的决定。虽然市面上有很多有关 C 语言的图书，但本书介绍 C 语言的方式最为合理，也让读者学习起来最为容易。本书的前五版都登上了畅销书排行榜，这一事实表明我们的观点得到了读者的认同。本书是按读者每天阅读一章的方式编写的。读者不需要有任何编程经验，但如果读者以前学习过其他语言（如 BASIC），学习起来将更快。另外，本书介绍的是 C 语言，而不针对任何编译器和计算机——读者使用的是 PC、Mac 还是 UNIX 系统将无关紧要。

本书包含一周的附加课程，旨在让读者对面向对象编程以及最流行的面向对象语言（C++、Java 和 C#）有个初步的了解。虽然这些章节无法全面地介绍这些主题，但将让您得以起步。

本书的特色

本书包含诸如语法、提示、注释、警告等内容，它们将给读者以启迪。“语法”介绍如何使用特定的 C 语言概念，对 C 语言命令和概念做了全面的阐述，并提供了范例。下面是一个“语法”范例：

printf()函数的语法如下：

```
#include <stdio.h>
printf(format-string[,arguments,...]);
```

printf()是一个函数，它接受一系列的参数(arguments)，其中每个参数用于给定格式化字符串中的一个转换说明符。该函数将格式化信息显示到标准输出设备（通常为屏幕）中。要使用 printf()，必须包含标准输入/输出头文件 stdio.h。

format-string 是必不可少的；但 arguments 是可选的。对于每一个参数，格式化字符串中都必须有一个转换说明符与之对应。格式化字符串可以包含转义序列。下面是几个调用 printf() 的范例及其输出：

范例 1:

```
#include <stdio.h>
int main( void )
{
    printf("This is an example of something printed! ");
}
```

输出:

```
This is an example of something printed!
```

范例 2:

```
printf("This prints a character, %c\na number, %d\na floating point,
%f", 'z', 123, 456.789 );
```

输出:

```
This prints a character, z
a number, 123
a floating point, 456.789
```

本书的另一个特色是“应该/不应该”，它告诉您应该做什么，不应该做什么。

应 该	不 应 该
应阅读本节余下的内容，它解释了每个课程最后的作业。	不要略过任何小测验和练习。如果能够完成当天的作业，则说明您已经为学习新的内容做好了准备。

本书中还有“注意”、“提示”和“警告”。“提示”提供了使用 C 语言的技巧和捷径；“注意”说明了一些特殊的细节，对 C 语言概念做了进一步的解释；“警告”帮助您避免发生一些潜在的问题。

本书通过大量的范例程序来说明 C 语言的特性和概念，让您能够在自己的程序中使用它们。对每个范例程序的讨论都由三部分组成：程序代码、用户输入和程序输出、对程序工作原理的分析。这些部分都由特殊的图标标识。

每章的最后都有“问与答”，其中包含与本章内容相关的常见问题及其答案。另外，每章的最后还有作业，其中包含小测验和练习。小测验检查读者对本章介绍的概念的理解程度。如果要检查自己的答案是否正确，或者对问题回答不上来，可参考附录 F 的答案。

仅仅阅读本书并不能学会 C 语言。要成为程序员，您必须编写程序。为此，我们在每一个小测验的后面提供了一组练习，建议您完成每一个练习。学习 C 语言的最佳方式是编写 C 语言代码。

我们认为，排错练习最有帮助。bug 是 C 中的一种程序错误，排错练习提供了一些代码，其中包含常见的问题（错误），您必须找出并更正这些错误。如果有困难，可参阅附录 F 提供的答案。

随着对 C 语言的介绍越来越深入，有些练习的答案将很长，而有些练习有多种解决方案。因此，对于最后几章中的有些练习，附录 F 没有提供答案。

本版所做的改进

任何事情都不是十全十美的，但我们尽可能追求完美。本书是《21 天学通 C 语言》的第 6 版。编写本书时，我们做了更大的努力，使其中的代码与更多的 C 语言编译器兼容。我们对本书做了多次审校，确保它在技术上极其准确。另外，根据读者的指正，我们更正了前几版中的很多错误。



注意: 我们在以下平台上对本书中的源代码进行了编译和测试: DOS、Windows、Macintosh、UNIX、Linux 和 OS/2。另外，本书以前版本的读者已经将这些代码用于了几乎所有支持 C 语言的平台中。

本版的另一项特色是 Type & Run 内容。本书共有 6 个 Type & Run，其中每个都包含一个简短的 C 语言程序。这些程序演示了 C 语言编程技术，同时能够完成一些有趣且有用的工作。您可以输入并运行这些程序。输入这些程序后，可以对其中的代码进行修改，看看能够使之完成其他哪些工作。Type & Run 就是提供给您进行试验的，希望您试验愉快！

附带光盘

为方便读者，附带光盘中包含了本书中所有的代码；另外，还有几个编译器，其中包括 Dev-C++——附

录 G 介绍了如何使用该编译器。

源代码的勘误和更新将在网站 www.samsublishing.com 上发布（如果有的话）。

本书的约定

本书使用不同的字体来区分 C 语言代码和正文以及指出一些重要的概念。源代码的字体为 Courier New；在程序的输出中，粗体表示用户输入的内容；占位符（应替换为实际的代码）则为斜体。

第一周课程

第 1 天课程 C 语言初步	2
1.1 C 语言简史	2
1.2 为何要使用 C 语言	2
1.3 编程前的准备工作	3
1.4 程序开发周期	4
1.4.1 创建源代码	4
1.4.2 编译源代码	4
1.4.3 链接以创建可执行文件	5
1.4.4 结束开发周期	6
1.5 第一个 C 语言程序	7
1.5.1 输入并编译 hello.c	7
1.6 总结	9
1.7 问与答	9
1.8 作业	10
1.8.1 小测验	10
1.8.2 练习	10
TYPE&RUN1 打印程序清单	12
第一个 TYPE&RUN	12
第 2 天课程 C 语言程序的组成部分	14
2.1 一个简短的 C 语言程序	14
2.2 程序的组成部分	15
2.2.1 main()函数 (第 8~23 行)	15
2.2.2 #include 编译指令 (第 2 行)	15
2.2.3 变量定义 (第 4 行)	15
2.2.4 函数原型 (第 6 行)	16
2.2.5 程序语句 (第 11、12、15、16、19、20、22 和 28 行)	16
2.2.6 函数定义 (第 26~29 行)	16
2.2.7 程序注释 (第 1、10、14、18 和 25 行)	16
2.2.8 使用花括号 (第 9、23、27 和 29 行)	17
2.2.9 运行程序	17

2.2.10 有关精度的说明	18
2.3 重温程序的组成部分	18
2.4 总结	20
2.5 问与答	20
2.6 作业	20
2.6.1 小测验	20
2.6.2 练习	21
第 3 天课程 存储信息: 变量和常量	23
3.1 计算机内存	23
3.2 使用变量存储信息	24
3.2.1 变量名	24
3.3 数值变量的类型	25
3.3.1 变量声明	27
3.3.2 typedef 关键字	27
3.3.3 初始化变量	28
3.4 常量	28
3.4.1 字面常量	28
3.4.2 符号常量	29
3.5 总结	31
3.6 问与答	32
3.7 作业	32
3.7.1 小测验	32
3.7.2 练习	32
第 4 天课程 语句、表达式和运算符	34
4.1 语句	34
4.1.1 空白对语句的影响	34
4.1.2 创建空语句	35
4.1.3 使用复合语句	35
4.2 表达式	35
4.2.1 简单表达式	35
4.2.2 复杂表达式	36
4.3 运算符	36
4.3.1 赋值运算符	36
4.3.2 数学运算符	37
4.3.3 运算符优先级和圆括号	40
4.3.4 子表达式的计算顺序	41
4.3.5 关系运算符	41
4.4 if 语句	42
4.4.1 else 子句	44
4.5 判断关系表达式	46
4.5.1 关系运算符的优先级	47
4.6 逻辑运算符	48

4.7 再谈 true/false 值	48
4.7.1 运算符的优先级	49
4.7.2 复合赋值运算符	50
4.7.3 条件运算符	50
4.7.4 逗号运算符	51
4.8 再谈运算符优先级	51
4.9 总结	52
4.10 问与答	52
4.11 作业	53
4.11.1 小测验	53
4.11.2 练习	53
TYPE&RUN2 猜数游戏	55
第 5 天课程 使用函数封装代码	57
5.1 函数是什么	57
5.1.1 函数的定义	57
5.1.2 函数的用法	57
5.2 函数的工作原理	59
5.3 函数和结构化编程	60
5.3.1 结构化编程的优点	60
5.3.2 规划结构化程序	60
5.3.3 从顶向下的方法	61
5.4 编写函数	61
5.4.1 函数头	62
5.4.2 函数的返回类型	62
5.4.3 函数名	62
5.4.4 参数列表	62
5.4.5 函数体	64
5.4.6 函数原型	67
5.5 将参数传递给函数	67
5.6 调用函数	68
5.6.1 递归	69
5.7 函数的位置	70
5.8 内联函数	70
5.9 总结	71
5.10 问与答	71
5.11 作业	72
5.11.1 小测验	72
5.11.2 练习	72
第 6 天课程 基本的程序流程控制	74
6.1 数组的基本知识	74
6.2 控制程序的执行	75

6.2.1 for 语句	75
6.2.2 嵌套 for 语句	79
6.2.3 while 语句	80
6.2.4 嵌套 while 语句	82
6.2.5 do...while 循环	84
6.3 嵌套循环	87
6.4 总结	88
6.5 问与答	88
6.6 作业	88
6.6.1 小测验	89
6.6.2 练习	89
第 7 天课程 信息读写基础	90
7.1 在屏幕上显示信息	90
7.1.1 printf() 函数	90
7.1.2 格式化字符串	90
7.1.3 转义序列	91
7.1.4 使用 puts() 显示消息	96
7.2 使用 scanf() 函数输入数值数据	97
7.3 三字符序列	100
7.4 总结	101
7.5 问与答	101
7.6 作业	102
7.6.1 小测验	102
7.6.2 练习	102
第一周复习	104

第二周课程

第 8 天课程 使用数值数组	110
8.1 数组是什么	110
8.1.1 一维数组	110
8.1.2 多维数组	113
8.2 命名和声明数组	114
8.2.1 初始化数组	115
8.2.2 初始化多维数组	116
8.2.3 数组的最大长度	119
8.3 总结	121
8.4 问与答	121
8.5 作业	121
8.5.1 小测验	121

8.5.2 练习	122
第9天课程 指针	123
9.1 指针是什么	123
9.1.1 计算机内存	123
9.1.2 创建指针	123
9.2 指针和简单变量	124
9.2.1 声明指针	124
9.2.2 初始化指针	124
9.2.3 使用指针	125
9.3 指针和变量类型	126
9.4 指针和数组	127
9.4.1 作为指针的数组名	128
9.4.2 数组元素的存储	128
9.4.3 指针算术	130
9.5 有关指针的注意事项	133
9.6 数组下标表示法和指针	133
9.7 将数组传递给函数	133
9.8 总结	137
9.9 问与答	137
9.10 作业	138
9.10.1 小测验	138
9.10.2 练习	138
TYPE&RUN3 让程序暂停	139
第10天课程 字符和字符串	141
10.1 char 数据类型	141
10.2 使用字符变量	141
10.3 使用字符串	144
10.3.1 字符数组	144
10.3.2 初始化字符数组	144
10.4 字符串和指针	144
10.5 不存储在数组中的字符串	145
10.5.1 编译时分配字符空间	145
10.5.2 malloc()函数	145
10.5.3 使用 malloc()函数	146
10.6 显示字符串和字符	148
10.6.1 puts()函数	149
10.6.2 printf()函数	149
10.7 从键盘读取字符串	150
10.7.1 使用 gets()函数输入字符串	150
10.7.2 使用 scanf()函数输入字符串	152
10.8 总结	154

10.9 问与答	154
10.10 作业	155
10.10.1 小测验	155
10.10.2 练习	156
第 11 天课程 结构、共用体和 TypeDef	157
11.1 简单结构	157
11.1.1 定义和声明结构	157
11.1.2 存取结构的成员	158
11.2 复杂结构	160
11.2.1 包含其他结构的结构	160
11.2.2 包含数组的结构	162
11.3 结构数组	164
11.4 初始化结构	167
11.5 结构和指针	168
11.5.1 将指针作为结构的成员	168
11.5.2 创建指向结构的指针	170
11.5.3 使用指针和结构数组	171
11.5.4 将结构作为参数传递给函数	173
11.6 共用体	174
11.6.1 定义、声明和初始化共用体	175
11.6.2 存取共用体的成员	175
11.7 使用 typedef 给结构创建别名	179
11.8 总结	179
11.9 问与答	179
11.10 作业	180
11.10.1 小测验	180
11.10.2 练习	180
第 12 天课程 变量作用域	182
12.1 作用域是什么	182
12.1.1 演示作用域	182
12.1.2 作用域为何重要	184
12.2 外部变量	184
12.2.1 外部变量的作用域	184
12.2.2 何时使用外部变量	184
12.2.3 extern 关键字	184
12.3 局部变量	185
12.3.1 静态变量和动态变量	186
12.3.2 函数参数的作用域	187
12.3.3 外部静态变量	188
12.3.4 寄存器变量	188
12.4 局部变量和 main()函数	189
12.5 应使用哪种存储类型	189

12.6 局部变量和代码块	189
12.7 总结	190
12.8 问与答	191
12.9 作业	191
12.9.1 小测验	191
12.9.2 练习	191
TYPE&RUN4 机密消息	194
第 13 天课程 高级程序流程控制	197
13.1 提早结束循环	197
13.1.1 break 语句	197
13.1.2 continue 语句	199
13.2 goto 语句	200
13.3 死循环	202
13.4 switch 语句	205
13.5 退出程序	212
13.5.1 exit()函数	212
13.6 在程序中执行操作系统命令	213
13.7 总结	214
13.8 问与答	215
13.9 作业	215
13.9.1 小测验	215
13.9.2 练习	215
第 14 天课程 操纵屏幕、打印机和键盘	217
14.1 流和C语言	217
14.1.1 何为程序的输入/输出	217
14.1.2 什么是流	217
14.1.3 文本流和二进制流	218
14.1.4 预定义的流	218
14.2 使用C语言的流函数	219
14.2.1 例子	219
14.3 读取键盘输入	220
14.3.1 字符输入	220
14.3.2 格式化输入	225
14.4 控制屏幕输出	231
14.4.1 使用 putchar()、putc()和 fputc()输出字符	231
14.4.2 使用 puts()和 fputs()输出字符串	233
14.4.3 使用 printf()和 fprintf()格式化输出	234
14.5 重定向输入/输出	238
14.5.1 重定向输入	239
14.6 何时使用 fprintf()	239
14.6.1 使用 stderr	240

14.7 总结	241
14.8 问与答	241
14.9 作业	242
14.9.1 小测验	242
14.9.2 练习	242
第二周复习	244

第三周课程

第 15 天课程 有关指针的高级主题	252
15.1 声明指向指针的指针	252
15.2 指针和多维数组	253
15.3 指针数组	259
15.3.1 复习字符串和指针	259
15.3.2 声明 char 类型指针数组	259
15.3.3 范例	261
15.4 函数指针	265
15.4.1 声明函数指针	265
15.4.2 初始化并使用函数指针	266
15.5 链表	273
15.5.1 有关链表的基本知识	273
15.5.2 使用链表	274
15.5.3 演示简单链表	278
15.5.4 实现链表	280
15.6 总结	286
15.7 问与答	286
15.8 作业	286
15.8.1 小测验	287
15.8.2 练习	287
第 16 天课程 使用磁盘文件	289
16.1 将流与磁盘文件关联起来	289
16.2 磁盘文件的类型	289
16.3 文件名	290
16.4 打开文件	290
16.5 读写文件数据	292
16.5.1 格式化文件输入/输出	293
16.5.2 字符输入/输出	296
16.5.3 直接文件输入/输出	297
16.6 文件缓冲技术: 关闭和刷新文件	300
16.7 顺序文件存取和随机文件存取	301

16.7.1	ftell()和rewind()函数	301
16.7.2	fseek()函数	303
16.8	检测文件尾	305
16.9	文件管理函数	307
16.9.1	删除文件	307
16.9.2	给文件重命名	308
16.9.3	复制文件	309
16.10	使用临时文件	311
16.11	总结	312
16.12	问与答	312
16.13	作业	313
16.13.1	小测验	313
16.13.2	练习	313
TYPE&RUN5 计算字符数		314
第 17 天课程 操纵字符串		318
17.1	确定字符串的长度	318
17.2	复制字符串	319
17.2.1	strcpy()函数	319
17.2.2	strncpy()函数	320
17.2.3	strdup()函数	321
17.3	拼接字符串	322
17.3.1	strcat()函数	322
17.3.2	strncat()函数	324
17.4	比较字符串	325
17.4.1	比较两个完整字符串	325
17.4.2	比较字符串的一部分	326
17.4.3	比较字符串时忽略大小写	328
17.5	查找字符串	328
17.5.1	strchr()函数	328
17.5.2	strrchr()函数	329
17.5.3	strcspn()函数	329
17.5.4	strspn()函数	330
17.5.5	strpbrk()函数	331
17.5.6	strstr()函数	331
17.6	字符串转换	332
17.7	其他字符串函数	333
17.7.1	strev()函数	333
17.7.2	strset()和strnset()函数	334
17.8	将字符串转换为数字	334
17.8.1	将字符串转换为 int	335
17.8.2	将字符串转换为 long 值	335
17.8.3	将字符串转换为 long long 值	335