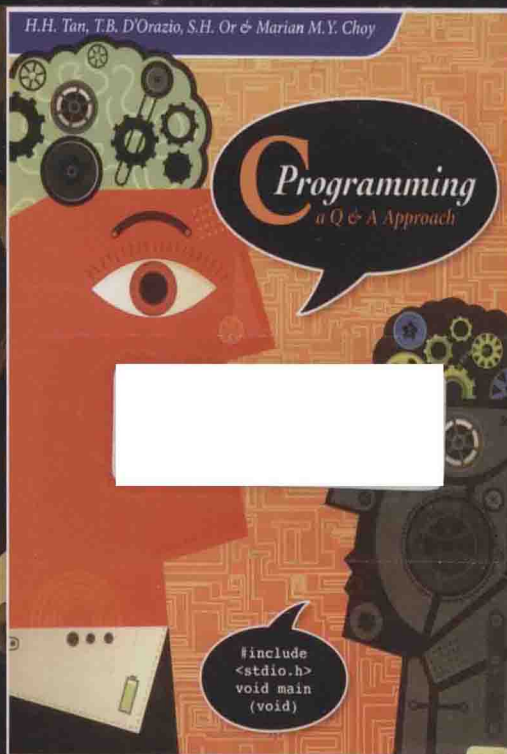


C语言程序设计

问题解答和实例解析方法

H. H. 塔恩 (H. H. Tan)
[美] T. B. 多拉齐奥 (T. B. D'Orazio) 著
柯兆恒 (S. H. Or)
玛丽安 M. Y. 周 (Marian M. Y. Choy)
赵岩 译

C Programming
a Q & A Approach



计 算 机 科 学 丛 书

C语言程序设计

问题解答和实例解析方法

H. H. 塔恩 (H. H. Tan)

[美] T. B. 多拉齐奥 (T. B. D'Orazio) 著
柯兆恒 (S. H. Or)

玛丽安 M. Y. 周 (Marian M. Y. Choy)

赵岩 译

C Programming
a Q & A Approach



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

C 语言程序设计:问题解答和实例解析方法 / (美) 塔恩 (H. H. Tan) 等著; 赵岩译. —北京: 机械工业出版社, 2016.7

(计算机科学丛书)

书名原文: C Programming: a Q & A Approach

ISBN 978-7-111-54334-3

I. C… II. ①塔… ②赵… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2016) 第 167847 号

本书版权登记号: 图字: 01-2012-4414

H. H. Tan, T. B. D'Orazio, S. H. Or, Marian M. Y. Choy: C Programming: a Q & A Approach (ISBN 978-007-131116-8).

Copyright © 2012 by McGraw-Hill Education.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education and China Machine Press. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2016 by McGraw-Hill Education and China Machine Press.

版权所有。未经出版人事先书面许可, 对本出版物的任何部分不得以任何方式或途径复制或传播, 包括但不限于复印、录制、录音, 或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和机械工业出版社合作出版。此版本经授权仅限在中华人民共和国境内(不包括香港、澳门特别行政区和台湾)销售。

版权 © 2016 由麦格劳-希尔(亚洲)教育出版公司与机械工业出版社所有。

本书封面贴有 McGraw-Hill Education 公司防伪标签, 无标签者不得销售。

本书以 C 语言作为工具, 通过大量实例, 详细介绍了基本程序设计的思想和技术。全书语言简练, 图示有助于理解, 围绕着读者通常关注的问题进行讲解, 强调问题的分析和讨论, 意在帮助读者认识程序设计的实质, 理解从问题到程序的思考过程。

本书适合作为高等院校计算机及相关专业第一门程序设计课程的教材, 也可供其他学习 C 程序设计的读者自学使用。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 和 静

印 刷: 三河市宏图印务有限公司

开 本: 185mm × 260mm 1/16

书 号: ISBN 978-7-111-54334-3

责任校对: 殷 虹

版 次: 2016 年 8 月第 1 版第 1 次印刷

印 张: 25.5

定 价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅肇划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

C 语言已经诞生了近半个世纪。在飞速发展的计算机领域，它完全称得上是一门古老的语言。虽然古老，但它和它的后继者 C++ 以及 Objective-C 一起形成的 C 语言家族，依然是目前计算机行业最流行的开发语言之一，尤其在一些要求速度和效率的应用（如操作系统、编译器等）中，始终占据着不可动摇的统治地位。

作为计算机科学与技术相关专业的学生，全面系统地学习 C 语言是十分必要的。如果不能很好地掌握 C 语言，将在今后学习操作系统和编译原理时遇到更大的困难。但是不可否认，C 语言由于在设计之初并不是一门教学语言，它的设计思想体现了很多底层的支持和对效率的考虑，如指针的概念、字符串的定义和操作、位操作等，这些内容对于刚刚接触计算机编程的学生，无论是学习还是掌握上，都具有一定的难度。

本书作者在长期的 C 语言教学中发现，作为一本实践性很强的学科，为了能让学生扎实地掌握 C 语言，必须要提供大量的编程实例，而不是在课堂上过多地讲解理论。通过实例学习编程虽好，但是当学生完成这些实例的时候，他们不可避免地会提出很多的问题。所以本书又将所有实例中经常出现的问题和解答与实例一并给出。通过实例，再配合问答，学生可以快速地掌握 C 语言中的重要知识点。通过在大学里的教学实验发现，利用本书作为教材，学生反映学习的难度降低了，同时对 C 语言的掌握也更加扎实了。

我出版过一本 C 语言的书籍，也从事过两年的 C 语言教学工作，所以深知 C 语言对于计算机专业的重要性以及 C 语言教学的难度。当我看到这种基于实例和问答方式进行组织的 C 语言教材时，感觉到本书的翻译和出版将对 C 语言教学做出一些非常有益的探索和尝试，所以毫不犹豫地接受了翻译的任务。

我在深刻理解全书内容的基础上力求准确，对于发现的笔误和印刷错误进行了更正。在本书的翻译过程中，得到了出版社和家人的支持和帮助，出版社的编辑对译稿提出了很多中肯的意见和建议。在此一并向他们表示感谢！

限于水平，译文中疏漏和错误难免，敬请读者批评指正。如有任何建议，请发送邮件至 zhaoyan.hrb@gmail.com。

译者

2016 年 4 月

新生经常发现阅读计算机语言书很困难，书写本书的目的就是为了解决这个困难。如果能使学生深入到书本中，激发他们的兴趣，并使得他们思考 C 语言的用法和含义，那么我们就可以把学习的过程变得简单且有趣。为此，我们使用了 Q&A 的方式。在这个过程中，学生经常问的问题也会激发读者的思考。通过直接并清晰地回答这些问题，我们把读者的注意力集中到 C 编程中重要的概念上。

我们也观察到很多计算机语言书很少有图。因为可视化的图形在教学中非常有用，所以我们努力使图示既准确又容易理解。对于程序执行的操作，这些图有利于澄清概念，加强学生对概念的理解。特别地，我们用三维的图来描述循环和判断结构，从而让学生可以很快地掌握程序的流程。我们相信这些图是对标准流程图的加强。我们也意识到对于很多学生来说，指针是最困难的部分。指针图示建立在包含变量名字、类型、地址和值的表格的基础之上，并且表格出现在文本解释的前面。本书中，我们使用表格来说明一个内存单元的信息是如何与另一个内存单元的信息联系起来的。

很多书包含大量的代码，但是并没有给出充分的解释，大部分新生不能也不愿意在没有解释的情况下独立地理解哪怕是很简单的代码。本书在代码中引导学生了解操作以及生成代码的过程，目的就是使学生意识到哪些地方需要额外的想法，以及掌握正确细节的重要性。

这种独特的方式已经受到用过本书草稿的学生的热烈欢迎。本书也被推荐给其他学生并询问他们的意见。当和其他书比较时，学生会优先选择我们的书。我们相信你在教学和学习过程中也会发现本书的价值。

本书组织

第 1 章介绍编程基础，假设学生除了会使用计算机进行简单的文字处理以外，没有其他的计算机知识。第 1 章介绍了编程语言的概念，描述了硬件、信息在内存中的存储方法、计算机语言、编译器和软件工程。本章的目标是使学生了解计算机的工作方式和软件设计背后的概念。

第 2 章到第 4 章讨论了过程语言的基本概念、基本语法和控制结构，也介绍了 C 库函数和它们的用法。第 5 章介绍了用户定义函数，强调了模块化和可重用代码的概念，简单介绍了指针，并将它用在一个传递地址的函数中。本章最后，介绍了使用 C 语言用户定义函数的效果。

第 6 章关注数值型数组。

第 7 章描述了字符串和指针。由于字符串使用地址进行管理，这一章也非常适合用来解释如何利用指针修改内存。第 8 章覆盖了 C 语言中的结构及其在生成链表、堆栈、队列及二叉树中的用法。另外，本章也介绍了大型程序设计，因为工程问题通常都很大。使用 C 特性来处理大型程序是公司招聘学生开发商用软件产品时的一项重要考量。

第 9 章是关于 C++ 语言的介绍。因为已经介绍过 C，所以更多地介绍 C++ 中面向对象

编程的核心概念。我们用简单的术语介绍了类、封装和多态。这一章有很多演示，简单的语言和丰富的演示为学生提供了很多使用 C++ 基本特性的背景知识。

大部分章节被分为两个部分——课程和应用程序。课程部分学习语法、格式和基本构造，应用程序部分演示课程中教授的知识如何用于解决实际问题，演示了开发的流程，目标是使学生能遵循结构化方法来开发自己的程序。

特点

1) 本书使用简单的问答方式，学生会发现这种方式比讲解的方式更加友好、更易于理解。这种方法下，作者能够发现学生经常问的问题并能简洁地回答这些问题。

2) 每一课都以一个样例程序开始：源代码并附有一些指示。学生根据指示观察代码的细节，从而了解 C 语言。下一步给出输出以及解释。解释环节给出一系列的问答以解释源代码做了什么。

3) 应用程序部分演示了 C 语言如何用于解决工程和计算机科学中的问题。我们详细地解释了它们。例子主要涉及程序设计、软件工程、模块化和生成可重用代码。

4) 给出大量的图来演示编程的概念。很多图都是独一无二的，能让学生快速地掌握概念。

5) 在应用程序部分描述了四步结构化方法（引入了字符串和更复杂的数据结构后变成了五步结构化方法）。方法包括生成结构流程图和数据流程图。

6) 应用程序部分也包括数值方法例子，这些例子用在把编程和数值方法结合起来的课程中。

7) 课程部分包含注释代码，以帮助学生理解程序的细节和流程，使学生关注代码并把代码中的重要部分高亮显示。

8) 我们意识到学生一般不会主动阅读多页代码，所以应用程序部分的每一段代码都只有 2 到 3 页，并有对应的解释。

9) 指针的概念很难理解。为了让学生理解指针，可视化图形是非常有用的。盒子中一个指针指向另外一个盒子，这种图是不够的。使用表格和网格状的内存草图，可以降低指针的神秘性。我们发现阅读本书后，学生能够轻松地理解指针的概念。

10) 应用程序部分后的练习可以用于实验课。教师可以让学生提前阅读特定的应用程序。上实验课时，可以指导学生做一些改动练习，后续的部分可以作为家庭作业。

11) 新生通常会在调试的时候遇到困难，因为他们对这个过程很陌生。新生也会感到很沮丧，因为他们必须要调试自己的第一个程序。为此，我们在第 1 章介绍了一个详细的调试例子。初学者也发现调试循环是很困难的，本书中关注循环并演示了循环中值是如何变化的。学生将学习如何追踪循环并发现错误。另外，初学者经常会犯的错误也在本书相应的位置指出。

12) 每课后面的判断题（有答案）可以让学生快速评价自己对基础知识的掌握程度。

13) 每章后面的应用练习可以作为家庭作业。

14) 本书中所有的程序都可以从 www.mheducation.asia/olc/cprogramming 获取[⊖]。学生可以修改并执行这些程序以理解它们是如何运行的。

⊖ 关于本书教辅资源，用书教师可登陆 www.mcgraw-hill.com.cn 申请。——编辑注

15) 第9章是有关C++的介绍, 不仅讨论了基础知识, 阅读完本章后, 学生还将学会使用面向对象编程的很多基本功能。

16) 很多应用程序介绍了数值方法。

如何使用本书

对于学生

在第1章, 你会理解什么可以保存在内存中, 编译器如何工作, 软件工程的步骤, 最重要的是, 编写自己的第一个C程序。其他章讨论了C语言编程。章节的课程都以一个简短的介绍开始, 指导你关注源代码中一些重要的知识点。然后你可以阅读源代码和其中的注解。你甚至可以运行代码并观察程序的行为。完成这些后, 确保你理解本课中主要的概念。之后阅读解释部分并完成判断题和简答题。如果做不好练习, 应重新学习本课以消除疑问。

掌握了一章的课程后, 开始学习应用程序部分, 其目的是演示编写程序的一般过程以及C的实际用处。你会发现, 当你写程序的时候会遇到很多应用程序中遇到的问题。在这一部分, 关注学习方法论及理解每一个程序的逻辑。记住, 编程中逻辑流是非常重要的。一个语句可能语法上正确但逻辑上却是错误的。掌握了每一个应用程序的来龙去脉, 会让你在写自己程序的时候更加自信。不要只是读, 要尝试每个程序, 修改并试验它。它会帮助你掌握在课程中学习过的内容, 进而解释程序的不同行为。利用这些知识完成教师布置的编程作业。

对于教师

作为一学期课程, 本书的目的是为学生在后续课程中掌握高级编程奠定基础, 例如C编程中带有C++的介绍, 推荐你按照顺序讲完所有的内容。但是, 按照不同的顺序讲解本书也是可以的。例如, 课程3.2(单个字符数据)可以在第7章的课程前讲述。同时, 如果需要的话, 课程8.7(生成头文件)、课程8.8(使用多个源文件及存储类别)、类似函数的宏和条件包含可以在第5章介绍。

你也可以将第7、8和9章的部分课程延后, 时间允许再讲解它们。例如, 课程7.9(指针符号与数组符号)可以延后到指针的高级话题(第8章的附加材料, 指向函数的指针和返回指针的函数, 通过www.mheducation.asia/olc/cprogramming获取^①)之前。

对于试图建立编程基础的一个学期课程, 我们推荐你讲解到课程7.8, 再加上课程7.10、8.1、8.2、8.3、8.4和8.5。

对于给学生一般的编程体验的短的课程, 如果只讲解前6章, 学生也会写出有价值且复杂的C程序。

本书提供了丰富的练习, 课程部分后有判断题和简答题。学生应该独立完成这些练习。课程后的一些简单程序可以留作作业。一个星期的时间学生足可以完成一个程序。

应用程序部分后面的修改练习可以用于实验课。学生应该在实验之前学习相关的应用程

① 关于本书教辅资源, 读者可向麦格劳-希尔教育出版集团北京代表处申请, 电话: 010-57997600/59575582, 电子邮件: instructorchina@mheducation.com。——编辑注

序。实验中，可以指导学生完成修改练习。一些练习比较容易，而另外一些很难，难的可以留作家庭作业。

大部分章末是应用练习。它们是本书中最有挑战性的练习，所以最适合留作家庭作业。根据不同的难度，需要 2 ~ 4 周的时间来完成它们。

另外，本书可以用作 ANSI C 的参考书，参考表格分散在本书正文中。

教师辅助材料

教师可在网站 www.mheducation.asia/olc/cprogramming[⊖] 获得以下补充材料。

- 解答手册
- 教学课件
- 测试库
- 附加练习
- 附加阅读材料

⊖ 关于本书教辅资源，用书教师可登陆 www.mcgraw-hill.com.cn 申请。——编辑注

感谢 McGraw-Hill 出版社的 Eric Munson 和 Holly Stark。感谢他们对本书的兴趣、支持、鼓励以及有见地的建议。与他们一起工作非常愉快。感谢 Byron Gottfried (BEST 系列的编辑), 感谢他的支持和宝贵的批评, 以及 McGraw-Hill 的 Alisa Watson 和她的产品团队, 使得本书版式优美。

我们有一些非常有想法的评论家。University of Houston 的 Betty Barr; University of Texas 的 Raymond Bell; Fayetteville State University 的 Tat W. Chan; Texas A&M University 的 Bart Childs; University of Minnesota 的 Chris J. Dovolis; Illinois State University 的 Janet Hartman; New Mexico State University 的 Elden W. Heiden; Purdue University 的 Elias Houstis; University of Minnesota 的 Joseph Konstan; University of Maryland 的 Jandelyn Plane; 还有 WPI 的 Matthew Ward, 他们都给出了非常有帮助的建议, 感谢他们的贡献。

我们的计算机生涯开始于 University of Michigan 和 U. C. Berkeley。感谢 J. M. Duncan, John Lysmer 和 Raymond Canale 教授 (我们第一年计算机课程的指导老师), 他们鼓励我们解决一些复杂的问题, 而由此带来的成就感和自信促使我们写这本书。

感谢 Suzanne Lacasse 和 Kaare Hoeg, 他们分别为 Norwegian Geotechnical Institute 的主管和前主管。他们对我们的自信以及在开发地理应用程序时给予的资助, 延展了计算机技术背景, 为进一步开发程序所需要的技能打下了坚实的基础。感谢 Hui Xian Liu (Institute of Mechanical Engineering 中国哈尔滨地区的前主管), 感谢他对 H.H. Tan 不断的支持、指导和鼓励。

最后要感谢我们的家庭。T.B. D'Orazio 的妻子 Elizabeth 虽然自己也很忙, 但还是抽出时间鼓励和支持我们。正是有了她, 辛苦的工作才有了乐趣。感谢 H.H. Tan 的妻子 Wei Huang, 她一直是一个热心、含蓄、聪明的女人。感谢 H.H. Tan 的女儿 Sijing Tan, 前四章的大部分图都是她绘制的。

H.H. Tan

T.B. D'Orazio

感谢 McGraw-Hill 的 Gerald Bok 和 Chris Cheung。Gerald 是这个项目的主要推动人，他处理了大量琐碎的事物，使得我们可以专注于本书的内容。Chris 负责沟通协作，也提供了很多帮助。也感谢 McGraw-Hill 的 Maureen Tan 的大力支持。

S.H. Or 要感谢香港中文大学的 John Lui 教授在担任计算机科学和工程系主任时给予的支持。同时也感谢自己的妻子 Pui-yee Lau，感谢她对孩子的细心照顾，使得自己可以专心于本书的编写。

Marian Choy 要感谢 Tim Lambert（在 University of New South Wales 她最受欢迎的讲师）让自己看到了编程的乐趣和创造性。也要感谢 Horace Ip 教授在香港城市大学给了她第一个教学的职位。同时也感谢香港大学工学院的 W.C. Chew 和 George L. Tham 教授，他们提供了免费的平台使自己可以探索和实验各种教学方式。最后，感谢所有学习过她的编程课的学生，感谢她的家庭、小组成员、朋友和同事，感谢他们一如既往的支持，以及对自己“固执地”追求高水平教学质量的包容。

S.H. Or

Marian M.Y. Choy

出版者的话
译者序
前言
致谢

第 1 章 编程基础	1
课程 1.1 编程语言	1
1.1.1 汇编语言	1
1.1.2 高级语言	1
课程 1.2 软件工程	3
1.2.1 自顶向下模块化设计	3
课程 1.3 C 语言、ANSI C 和 C 编译器	5
1.3.1 C 和 ANSI C	5
1.3.2 程序开发	5
课程 1.4 利用位表示字符、符号、整型 数、实型数、地址和指令	7
1.4.1 字符和符号	7
1.4.2 整型数	8
1.4.3 实型数	9
1.4.4 十六进制和八进制表示	9
课程 1.5 关于本书及如何充分利用 本书	10
1.5.1 课程	10
1.5.2 应用程序	11
课程 1.6 基本结构	12
课程 1.7 格式化输出	16
课程 1.8 其他转义字符	18
课程 1.9 基本调试	20
本章回顾	22
第 2 章 变量、算术表达式和输入输出	24
课程 2.1 变量：命名、声明、赋值和 打印值	24
课程 2.2 算术运算符和表达式	30
课程 2.3 从键盘输入数据	33

课程 2.4 常量宏及打印变量值的进一步 讨论	37
课程 2.5 混合类型的运算、复合赋值、 运算符优先级和类型转换	43
本章回顾	52
第 3 章 C 语言基础：数学函数和字符 文件输入输出	53
课程 3.1 数学库函数	53
课程 3.2 单个字符数据	57
课程 3.3 从文件读入数据	67
课程 3.4 输出到文件	72
应用程序 3.1 面积计算——复合运算符 和程序开发	74
应用练习	77
本章回顾	79
第 4 章 初级决策和循环	80
课程 4.1 if 控制结构和关系表达式	80
课程 4.2 简单 if-else 控制结构	84
课程 4.3 嵌套 if-else 控制结构	87
课程 4.4 逻辑表达式	90
课程 4.5 逻辑运算符的优先级	92
课程 4.6 switch 和 if-else-if 控制结构	96
课程 4.7 while 循环 (1)	102
课程 4.8 while 循环 (2)	105
课程 4.9 do-while 循环	107
课程 4.10 简单 for 循环	109
课程 4.11 嵌套 for 循环	112
应用程序 4.1 梁交叉——if-else 控制 结构	116
应用程序 4.2 面积计算——for 循环	118
应用程序 4.3 温度单位转换——for 循环	120
应用程序 4.4 温度单位转换——循环和 if-else 控制结构	121
应用程序 4.5 仿真	123

应用程序 4.6 工程经济学——嵌套 for 循环	124	应用练习	215
应用程序 4.7 解二次方程——if-else 控制结构 (数值方法例子)	126	本章回顾	219
应用练习	128	第 7 章 字符串和指针	220
本章回顾	131	课程 7.1 声明、初始化和输出字符串及理解内存布局	222
第 5 章 函数	132	课程 7.2 确定字符串和字符信息及使用 printf	229
课程 5.1 不返回值的函数	134	课程 7.3 二维字符数组	234
课程 5.2 返回一个值的函数	142	课程 7.4 从键盘和文件读入字符串	238
课程 5.3 作用域和传值给函数的机制	146	课程 7.5 指针变量与数组变量	245
课程 5.4 返回多个值的函数	151	课程 7.6 在声明中初始化	251
课程 5.5 从函数返回多个值的机制——地址和指针变量	153	课程 7.7 将字符串传入用户自定义函数	256
应用程序 5.1 使用带有复杂循环的函数处理网格 (逻辑例子)	159	课程 7.8 标准字符串函数	261
应用程序 5.2 模块化程序设计: 平行四边形面积和平行六面体体积 (数值方法例子)	164	课程 7.9 指针符号与数组符号	272
应用练习	167	课程 7.10 动态内存分配	279
本章回顾	172	应用程序 7.1 管流速、检查输入数据及模块化设计	285
第 6 章 数值数组	173	应用程序 7.2 地震轶事报告分析、字符串操作和动态内存分配	294
课程 6.1 一维数组和打印数组元素介绍	174	应用练习	305
课程 6.2 数组初始化	178	本章回顾	308
课程 6.3 基本数组输入输出	181	第 8 章 结构和大型程序设计	309
课程 6.4 多维数组	185	课程 8.1 结构	310
课程 6.5 函数和数组	192	课程 8.2 结构成员	316
课程 6.6 冒泡排序和最大交换排序	197	课程 8.3 指向结构的指针	318
应用程序 6.1 将 16 个 1 位加法器组成 1 个 16 位加法器	202	课程 8.4 结构和函数	321
应用程序 6.2 浪高的平均值和中位数 (数值方法例子)	205	课程 8.5 结构数组	322
应用程序 6.3 矩阵-向量乘法 (数值方法例子)	209	课程 8.6 带一个递归调用的函数	324
应用程序 6.4 搜索和文件压缩	212	课程 8.7 生成头文件	329
		课程 8.8 使用多个源文件及存储类别	331
		课程 8.9 位操作	334
		应用程序 8.1 排序——快速排序算法	342
		本章回顾	350

第9章 C++ 介绍	351	课程 9.7 带有数据和函数成员类及封装	369
课程 9.1 C++ 注释和基本输入输出流	351	课程 9.8 构造函数和析构函数	375
课程 9.2 格式操纵符及格式化输出	354	课程 9.9 继承	379
课程 9.3 函数重载	357	应用程序 9.1 电子电路	385
课程 9.4 默认函数参数	360	应用练习	389
课程 9.5 内联函数和变量声明的位置	363	附录 A ASCII 码	391
课程 9.6 C++ 类和只有数据成员的对象	365	附录 B ASCII 码描述	392

编程基础

本章目标

结束本章的学习后，你将可以：

- 编写一个完整的 C 程序。
- 使用 C 语句给出格式化的输出。

用计算机编程并不是一个简单的任务，但是那种伴随着自己程序成功运行的喜悦感也许是对程序员最大的奖励。在本章中你将学习如何编写一个简单的 C 程序，在屏幕上输出一些单词、数字和键盘符号。本章所介绍的示例程序本身的实际用途并不是很强，但是它们对学习 C 程序设计是很有帮助的。

在介绍第一个程序前，我们先简单了解一下编程的过程。

课程 1.1 编程语言

为什么要学习编程？这个问题需要详尽的阐述和回答。假设我们想控制计算机，第一件要做的事就是与这个可编程的机器进行交流。机器语言是计算机能够理解并运行的唯一语言，也就是说，它是唯一能够控制计算机的语言。机器语言由二进制的指令构成，即一系列 0 和 1 组成的代码，并只针对某种特定的处理器，例如 Intel 的 Pentium 系列处理器。计算机执行的每一步都需要用这些指令书写。因为机器语言非常繁琐，所以大部分程序都是先用其他的语言编写，然后被翻译成机器代码。

1.1.1 汇编语言

汇编语言被认为是比机器语言高一个层次的语言。在汇编语言中，所有的指令和对应的机器语言需要一一映射。例如，对某个特定的处理器来说，机器语言中的加法指令是“10010101”，而在汇编语言中，对应的指令代码是“ADD”。汇编语言中的指令不再是二进制码，而是英文单词，这种代码样式对人类来说更可读，这是汇编语言相对机器语言来说一个主要的优点。英文单词可以被语言翻译程序翻译成对应的机器语言。我们可以想象把英文单词直接转换成对应的二进制码并不是一件很困难的事。

虽然汇编语言比机器语言更好用，但是使用汇编语言的最大问题是它要求程序员对硬件有非常全面的了解。另外，为了完成诸如一些简单的计算和输出信息的任务，程序员需要书写大量的汇编语言代码。

1.1.2 高级语言

高级语言是为了进一步简化程序员需要书写的命令而开发的。例如，在机器语言或者汇编语言中，为了把两个数加到一起，我们需要在计算机的内部采取一系列的步骤，把信息从一个内存单元转移到另外一个内存单元中。而对人类来说，一个更简单的写法就是“a+b”。当这种语言被翻译后，用来把两个数加起来的一系列机器语言指令已经被写出并保存在内存

中。与机器语言不同，高级语言允许程序员不用关心运行程序的机器的内部设计情况。

高级语言需要满足一些规则使得自己能够被正确地翻译成机器语言。这些语言被设计用来简化可以解决特定问题的编程。例如，有些语言被开发用来编写能够解决科学计算问题的程序，而有些用来处理商业事务。

编程语言（见表 1-1）可以被分成 4 种类型：

- 1) 过程（或指令）语言
- 2) 函数语言
- 3) 声明语言
- 4) 面向对象语言

目前，还不需要详细说明这四种语言之间的区别。但是需要指出的是，C 语言是一个面向过程的语言。就像这个名字所暗示的，面向过程的语言需要程序员安排一系列的过程来解决问题。正如在本章的应用程序部分你将看到的：针对特定的问题，首先关注于开发一些过程，然后编写一个能够执行这些过程的程序。

在表 1-1 中还需要注意 C++ 是一个面向对象的语言。抛开区别不说，C 是 C++ 的一个子集；这也就是说，本书中你学到的任何关于 C 语言的知识，在将来学习 C++ 的时候都会用得到。

表 1-1 高级语言总结

语言名称	语言类型	开发时间	语言名称	语言类型	开发时间
Fortran	过程语言	20 世纪 50 年代中期	Smalltalk	面向对象语言	20 世纪 70 年代中期
Basic	过程语言	20 世纪 60 年代中期	Pascal	过程语言	20 世纪 70 年代早期
Lisp	函数语言	20 世纪 50 年代末期	C	过程语言	20 世纪 70 年代中期
Prolog	声明语言	20 世纪 70 年代早期	C++	面向对象语言	20 世纪 80 年代中期
Ada	过程语言	20 世纪 70 年代中期			

语言翻译器

语言翻译器是一类程序，负责把汇编语言或高级语言编写的指令转换成对应的机器语言（也叫目标码）。C 语言编写的计算机程序需要用语言翻译器编译成机器语言。有三类语言翻译器：汇编器、编译器、解释器。

汇编器负责把用汇编语言写成的程序转换成目标码。因为汇编语言和机器语言非常类似，所以汇编器比解释器和编译器更简单。

编译器负责把整个用高级语言书写的程序转换成机器指令。在转换的过程中，高级语言代码中的错误可以被编译器检测到。编译器会查找那些违反了语言制定规则的问题，诸如不正确的标点符号或者冲突的声明；但是编译器并不负责检测出所有的错误，例如逻辑错误。通常情况下，入门的程序员通常会错误地假设经过编译器正确编译过的程序就一定会给出正确的结果。一个编译器编译时并没有给出任何错误信息的程序，就像一个没有任何语法错误的英文文章。它满足特定的规则，但是并不一定就有意义。

你将会一直用 C 语言编译器把 C 语言编写的指令翻译成机器语言代码。编译器会查找代码中的语法错误和不一致的地方，并将这些结果描述给你。在这种情况下，程序需要经历一个叫做调试（debugging）的过程，也就是说，修改程序中的错误以满足编译器的要求。然后需要运行并检验程序以确保它完成任务。如果程序并没有正确地运行，你需要修改、增加或删除一些 C 语言指令，重新编译整个程序并运行它。你需要一直重复调试的过程，直到

从程序得到令人满意的结果。

市场上有很多C语言的编译器，你的大学可能会提供一个编译器让你使用。你的老师会教会你如何使用这个编译器。你也可以自己从软件商店购买一个C语言编译器，或者使用互联网上的一些免费的编译器。这些编译器附带的指令会告诉你如何把C代码翻译成机器码以及如何正确使用编译器的其他特性。

翻译器也用于高级语言。与编译器不同，翻译器依次翻译并执行指令。换句话说，翻译器取出高级语言书写的一个指令，将它转换成机器语言并运行它。然后翻译器处理下一个指令，并一直重复这个过程直到所有的指令被执行完毕。

概念回顾

1) 与计算机交流的语言可以分为：

- 高级语言——这种编程语言更加面向人，程序员不需要了解计算机内部的结构就可以使用它进行编程。
- 低级语言——编程过程以及把程序在计算机间进行移植更加困难。

2) C语言可以被认为是一种高级语言。

3) 为了能在计算机上运行C程序，需要一个编译器把C语言程序转换成可执行的机器语言程序。

课程 1.2 软件工程

软件工程是一个用来描述软件开发过程的术语。它表明软件并不是被任意开发出来的。与此相反，软件的特性必须被仔细考虑、计划、构造和检测。这个术语反映了构造软件和构造硬件之间的类比性，也反映了建造软件和建造任何其他传统工程之间的类比性。

软件开发是一个精心计划的过程，软件的功能首先被定义。先要做出一个初始的计划，并且咨询所有软件开发相关方（用户、拥有者、程序员及其他）的建议。然后不断地修改，每个单独组件的设计也需要涉及。当软件进行整体的组装和测试时，需要计划和实施修改。软件的文档也要仔细地维护以使用户可以高效地使用以及方便日后的修改。同时，需要预测每一个开发步骤的时间和花费使得工程可按计划完成。

软件设计和开发的很多步骤在图 1-1 中给出。注意图中的“循环”，循环就是重复的步骤，也就是说，测试和修改的步骤。这些是非常重要的，因为软件会持续地经历修改和测试过程。

本文中，你会学习如何计划一个初始的程序结构，书写C语言的代码，遵循模块设计方案，测试程序，修改并书写文档。这些并没有包含所有的软件开发步骤，但是它们足够你计划和构造有用的C程序。

1.2.1 自顶向下模块化设计

自顶向下的设计方法从定义软件的主要功能开始，然后开发次要功能。每一个组成部分都更简单，并且可以被单独设计。所有的部分需要正确地组合在一起。每一个单独设计的组成部分都可以被认为是一个模块。

在C语言中，模块被称为函数。这些函数主要分为两种类型：库函数和用户定义函数。库函数是一些已经被开发出来并包含在编译器中的模块。它们执行一些标准数学的操作，诸