

C/C++ 实践进阶之道

写给程序员看的编程书

陈黎娟◎编著

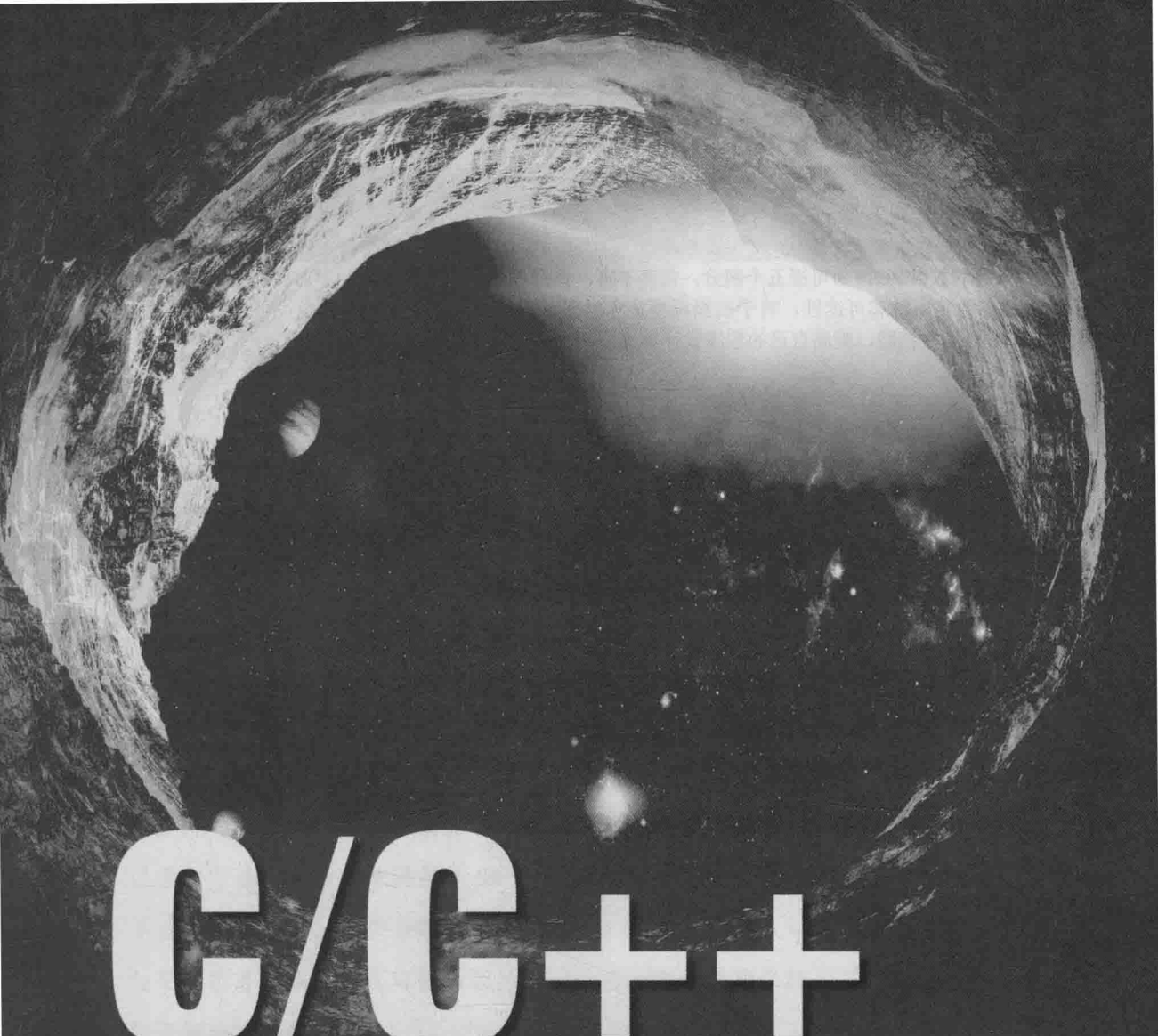
浸透作者 **十年** 开发经验

珍贵心得与踩坑经历娓娓道来，助力读者扎实进阶提升
源代码、精彩开发视频及电子教程组成资料包，倾囊奉送



超值资料包
扫码下载

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



C/C++

实践进阶之道

常州大学图书馆
藏书章

写给程序员看的编程书

陈黎娟◎编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

这是一本 C/C++ 语言应用能力进阶的图书，总结了一个典型的学习者第 101 个小时开始的学习路线和心得，你值得拥有。

本书内容包括夯实 C 语言核心基础、提升你的编程功力、积累专业程序员开发经验、理解 C++ 精髓和以 C++ 为母语五个部分，内容丰富，集开发技巧、成长经验和学习资料于一体，具有很高的实用性和可读性，对于初级程序员和学习编程语言的大学生大有裨益，可以帮助其掌握 C/C++ 语言精髓，提高自己的程序设计水平。

图书在版编目 (CIP) 数据

C/C++ 实践进阶之道：写给程序员看的编程书/陈黎娟
编著. —北京：中国铁道出版社，2018.5
ISBN 978-7-113-23758-5

I. ①C… II. ①陈… III. ①C 语言-程序设计②C++ 语言-程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 214113 号

书 名：C/C++ 实践进阶之道：写给程序员看的编程书
作 者：陈黎娟 编著

责任编辑：荆 波
责任印制：赵星辰

读者热线电话：(010) 63560056
封面设计：MX DESIGN
STUDIO

出版发行：中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)
印 刷：中国铁道出版社印刷厂
版 次：2018 年 5 月第 1 版 2018 年 5 月第 1 次印刷
开 本：787 mm×1 092 mm 1/16 印张：29.75 字数：724 千
书 号：ISBN 978-7-113-23758-5
定 价：79.00 元

版权所有 侵权必究

凡购买铁道版图书，如有印制质量问题，请与本社读者服务部联系调换。电话：(010) 51873174

打击盗版举报电话：(010) 51873659

作者序

希望能给你带来帮助

十多年前，我大学毕业。

十多年来，我一直从事开发工作。主要的开发工具是：C/C++语言。当然，还会使用一些其他的开发语言 and 平台。

十多年了，也积累了一些东西，决定抽空把自己技术上成长的路总结一下，希望能给你带来一定的帮助。

在大学的时候，我只不过是一个想努力学好每门功课，争取拿到奖学金的学生。

C语言对我来说，只不过是其中的一门功课，对功课的态度我都是一样的。

记得我的这门功课的考试得了98分。

不过，现在看来，这个分数毫无意义，只不过却跨出了认识这门语言的第一步。

如果说重要的课程，反而是后面的一门算法和数据结构课程，工作时间越久，越意识到这门课程的重要性。

算法和数据结构课程之后，又一次接触C语言是在毕业设计时。

毕业设计对我来说，有两个重大的意义：

一是编写了一个“大”程序，让我意识到要组织好代码，还得具有一些工程化的知识经验技巧规范。

二是编写了这么多代码，C语言让我使用得更加熟练。

不过，当你知道得越多，接触到未知的东西就会越多，那个时候才明白，面对整个C类开发而言，我的C语言成绩应该只有2分。

毕业后，我经历了很多比大学毕业设计更加庞大的项目。

非常幸运的是，这些项目都是由浅入深，逐步走入我的工作中，这逐渐给了我成长的空间。

第一个阶段：这些项目，都是嵌入式在 ROM 中的小程序，它们让我学会了算法平衡，学会了设计，学会了调试……

第二个阶段：特点是这些项目都引入了嵌入式开发系统。项目的规模很大，系统的规模也相当大，需要很大的团队来支持、完成。这些项目让我达到了另外的台阶，基本上不再是个人独立完成项目，开始积累项目开发的经验……

总结一下，你会发现这些经验非常可贵，我要是早早就知道这些，工作中可能就会少犯很多低级错误，还有莫名的焦躁。

……

我平时也有记录工作笔记的习惯，这些经验都散布在我厚厚的几个笔记本中。

一直想，等我有空了，把这些总结一下，应该是非常有意思的事情。不过一直也没有时间。

直到有一天，和我一个同学聊天，他说起来目前的大学生就业形势。我知道很多公司都缺乏开发人才，而且能开出的工资远远高于网络上公布的平均数字。而这些工作的基本要求，就是熟练掌握 C 语言。实际上，我们读大学的时候，理工科大学学生学习一门编程语言是一个基本要求，我想现在也是，所以，我觉得这个要求并不算高。大学生缺的，只是熟练程度和一些基本经验。

所以，我决定每周总结一点。希望能在你找工作的时候有所帮助，希望能对你的成长有一些帮助。

Emily
2018 年 3 月

C/C++的数组是从0开始编号，我们也从第0章开始吧！

作为职业程序员，在生活和网络中，学弟学妹们问得最多的两个问题是：收入如何？你是如何成为专业程序员的？对于第一个问题，我一般回答，收入还可以，关键是有时的成就感非常让人满足。而第二个问题还真不好回答。我只能说读书的时候，就有点儿喜欢编程，就业的时候，发现程序员职位的薪水比较高，工作了几年，就逐渐成了职业程序员了。要详细地说，基本上需要一本书的内容。这也是本书的缘起。

读者：理工科大学学生

有一个现象，在职业程序员中，大学是计算机专业的并不多。这说明，一方面，计算机应用的迅猛发展，使得计算机应用人才缺乏；另一方面，不少理工科专业和计算机专业天然接近，成为职业程序员也不是一件很难的事情。

事实上也是如此，大多数理工类专业开设了不少程序设计的相关课程，只需要有一定的兴趣爱好，再加上一定的训练，可以比较轻松地成为专业程序员。因为计算机在各行各业的应用，对于理工科专业的学生来说，编程能力也是一个非常重要的技能。

所以，本书将理工科大学在校生和刚刚进入职场的新人，作为读者对象，希望会对这些人有所帮助。

定位：你的第二本程序设计图书

几乎所有理工类大学都会开设程序设计语言课程。每年也会有数百万人报名参加计算机等级的程序设计类考试。这是成功的第一步。但是，会基本语法，不一定会编写程序，不一定能运用到实际中去解决问题。

本书定位于理工科大学学生的第二本程序设计图书，在已有的程序设计语言的基础上逐步提高，具备专业程序员的基础。花一点时间阅读本书，你就可以多拥有一项能力。从学生到职业程序员，在工作和生活等多方面都要有一定的转型和适应期。本书出版的最重要目的是帮助读者完成这个转型，帮助读者跨越几个技术学习上的台阶。

熟练掌握 C/C++的意义

掌握 C/C++的好处如下：

(1) C 语言和 C++基本上是使用量第一的开发语言，C/C++是现代程序设计之母，C#和 Java 等都是在 C++的基础上演化而来的，因此掌握了 C/C++，对学习 Java 和 C#会很有帮助。

(2) 在 C/C++ 的提高学习中, 会涉及计算机的数据结构、操作系统等知识, 会提高自己解决问题的综合能力。程序设计的学习领域很多, 范围非常广, 新概念、新知识层出不穷。但只要你掌握的知识结构合理, 综合能力高, 学习这些新东西将非常容易。

(3) 熟练掌握 C/C++, 是很多单位招聘的基本要求, 大多数理工科专业的在校大学生已经学习了 C 语言的基本语法, 或者已经有了一定的编程基础, 只需要再花一定的时间巩固掌握即可。

成为专业程序员的台阶

我们都知道, 从学生到职场人士, 有几个台阶需要跨越。回顾一下一个 C/C++ 程序员的成长过程:

第一个台阶, C 语言入门。简单来说就是能通过大学的 C 语言考试, 或者通过计算机等级考试的二级 C。

第二个台阶, 熟悉库函数。C 语言的不少功能是通过库来实现的, 学习库函数实际上是熟悉 C 语言的重要阶段。

第三个台阶, 能编写比较大一点儿的程序。通过编写千行内的程序, 可以积累函数划分、调试经验, 慢慢对 C/C++ 语言会用得非常熟练。

第四个台阶, 能自己编写一些小的工具。这种训练能积累解决问题的方法。

第五个台阶, 建立 Project 的概念, 能把程序分解成几个代码文件。

第六个台阶, 洞悉程序开发的真相。理解算法、数据结构、解决问题的方法, 领域类知识等, 这些比语法更重要。

第七个台阶, 掌握宏、指针、内存管理、static 和 const 等专业 C/C++ 程序经常需要的关键用法。

第八个台阶, 掌握 C++ 的精髓。C++ 虽然内容丰富, Core 和 C 语言也不大。这些内容, 也是程序设计语言的精髓。

在跨越了这八个台阶之后, 只需要再经历一些项目开发的强化训练, 即可成为一个合格的 C/C++ 程序员。

掌握 C/C++ 的学习地图

结合十多年的 C/C++ 学习和成长经验, 结合当前就业的需要, 我们勾勒出一个在校大学生或职场新人的程序设计学习地图。

如何学习才能快速掌握 C/C++ 呢? 程序员积累了丰富的经验并总结出了下面的学习经验。只要你跟随学习方案, 每周抽出 3 个独立时间段进行上机实践, 每次至少 2 个小时。一共大约需要 240 个小时的上机训练, 即大约 40 周的自我训练。最多 2 个学期, 你就能获得一个质的飞跃。再辅以其他适当的训练, 即可成为一名合格的程序员。

学习阶段	学习内容和达到目标	上机实践
入门	基础语法练习，掌握常用数据类型的定义，三种程序结构，函数，数组，指针，结构体和文件等的用法	30 小时
提高	熟悉库函数： 掌握标准库函数的主要头文件和分类，至少掌握常用的 10 多个库函数	10 小时
	逐步编写大一点的程序： 至少要编写几个几百行代码的程序，学会自己组织编写函数，分配各个函数的功能模块	20 小时
	自己编写小工具： 结合一些典型应用，自己尝试编写一些小的工具软件，这是提高编程水平的一个窍门。也是编程的乐趣所在。一般来说，自己能独立编写三个小工具之后，即可慢慢体会到编程的奥秘，后面提高就快了。这是学习的一个关键点，也可以说是转折点	30 小时
	学会组织代码，懂得 Project 的意义： 每一个开发工具都提供了一个 Project 的代码组织方式，练习自己做一个 Project，掌握之后，即可说是掌握了所有开发工具的代码组织方式	10 小时
	理解文件格式，数据结构和视图表现三者之间的关系： 任何一个专业应用程序，磁盘上的文件格式，内存中的数据结构和屏幕上的界面表现，都是非常重要的三个环节，如果能理解这三者之间的关系，并应用在程序开发中，可以说已经具备了专业程序员的基础	20 小时
熟练	学习基本数据结构，掌握经典算法	30 小时
	接触语言的细枝末节，吸收积累代码编写经验： 对 C/C++ 来说，就是宏，指针和内存管理，这是三大难点。 是否会准确使用 const, static 等关键字语法，也是程序员和普通学生的区别	20 小时
	学习某一个库，编写一些综合应用： 对程序员来说，快速学习库、应用库，这是基本技能，通过一个库的学习和练习，可以积累很多经验和开发细节	30 小时
	掌握类和对象的基本语法，理解封装和虚函数等概念	10 小时
掌握 C/C++ 精华	理解多态的优点，会用 C/C++ 编写一些小的工具	20 小时
	了解 C/C++ 精髓之外的其他语法形式，用到的时候能拿来即用，理解设计模式。重构等现代开发概念	10 小时

根据学习地图和学习经验，我们设计了本书的内容。全书共分为 5 篇，下面分别介绍这五篇的内容安排。

第 1 篇 夯实 C 语言核心基础

本书定位的是读者的第二本程序设计类图书。所以，读者可能有以下三种情况：

- (1) 已经学习了一门《C 程序设计》之类的课程。第 1 篇，可被视作为一个 C 程序设计的浓缩，帮助你回顾了解 C 语言最核心的内容。
- (2) 正在学习《C 程序设计》的课程。第 1 篇可以作为教学辅导图书，一边学习教材，一边阅读本书，时刻提醒读者注意入门之初别纠缠于语法细节，起到帮助读者纠正学习方

向的作用。

(3) 学习了一门程序设计语言，是 VB 之类的非 C 程序设计语言。第 1 篇可以作为快速学习 C 语言的教材。当然，这些内容还远远不够，我们在网上提供了一个免费的 C 语言学习教材。

总的来说，第 1 篇主要是配合读者正在学习的 C 语言教材，帮助读者掌握 C 语言核心的语法。同时，帮助读者建立重要的程序设计概念。

第 2 篇 提升你的编程功力

按照学习地图，提供全书重要的学习概念和学习内容。

首先安排库函数的学习，通过学习库函数的应用，一方面熟悉语法，另一方面，建立接口的观念，进一步通过自己编写个别库函数的训练，可以较快的提升编程能力。

其次是逐步编写大软件的项目和办法。教会读者如何编写自己的小工具软件，让读者在趣味中学习。

最后是让读者自己掌握软件运行时的数据结构、硬盘上的文件数据和用户看到的界面之间的奥秘。

当读者明白了这些奥秘后，就会知道自己需要掌握的知识、算法和数据结构，解决问题的方法。接下来，本书通过其他章节，将读者引入这些领域中，让读者得到初步的修炼。

可以说，这部分内容是本书的及格线。即使读者没有掌握本书的其他内容，只要能理解这部分内容，对读者提升对计算机的认识将会有非常大的帮助。

有了上面的基础，可以进入熟练掌握一些关键语法的阶段，这就是 static、const、指针、内存管理等。我们对这些内容会比较详细的讲解。要知道，这些内容一般都是存在于多本图书之中，而这些内容，又是 C/C++ 程序员必须掌握的语法内容，面试常考，偏偏理解起来又有一些难度。所以，编写时将其放在了熟悉语法，编写了几个小工具之后，作为提高训练之用。

第 3 篇 积累专业程序员的开发经验

掌握了 C/C++ 之后，也许会成为嵌入式开发程序员。毕竟，在信息时代，这样的工作岗位会越来越多。

为此，我们也花了一定的篇幅来介绍在跨越第一到第七个台阶之后的程序员。掌握哪些知识，可以成为一个合格的嵌入式程序员。要知道，嵌入式开发专业方面的人才需求极大，数学、物理、机电、仪器、计算机、电子等相关专业的学生，成为这个行业的主要人才来源，这就相当于给这部分同学，提供了一个就业出口。

第 4 篇 C++ 和类——面向对象的世界观

C++ 领域有很多经典名著，但这些图书都有一个统一的缺点：太厚。给初学者以恐惧感，不太容易看得完。

C++确实博大精深，编程模式多，涉及面广。比如，引入类和对象的概念之后，和前面的知识一交叉，可以有类和指针交叉：对象的指针，对象的内存分配等；类和数组交叉：对象数组；类和函数的交叉：方法和函数的区别，对象作为函数的参数等。引入任何一个新的概念，会引发连锁反应。更何况 C++ 引入的内容越来越多。因此很少有人敢称他掌握了 C++ 的全貌。

在实践中，我们会发现，经常用到的只是类、封装、继承、虚函数和多态等几个有限的 C++ 精华，这基本上是所有面向对象开发语言的最小核心子集。

接下来，我们以抽丝剥茧的方式，把这些 C++ Core 内容抽取出来，重点讲解，作为读者学习其他 C++ 程序设计教材的补充。

本篇内容和 C++ 教材结合起来学习是比较好的选择。一个全面，一个重点；一个重在语法细节的阐述，一个重在关键要点的把握；一个重在从语言学习的角度编写，一个从实践需要角度侧面叙述。一厚一薄，互相配合，学习起来更容易。

第 5 篇 以 C++ 为母语

并不是每一个读者，将来都会成为一个 C/C++ 程序员。

大多数现代的程序设计语言都和 C++ 有一定的亲缘关系，Delphi、Java、C# 这些先后出现的面向对象的开发语言，在 C++ 的基础上，都有一些自己的独有的特点。对学习来说，在 C/C++ 上的学习投资，完全可以用到这些语言的学习上。你的 C++ 功力越深厚，学习这些新的开发语言的时间就越短。

本篇简单讲述如何在已经掌握的知识基础上，快速学习这些带有面向对象开发特性的程序设计语言。强化每个现代人都必须掌握的学习能力。

后续学习与提高

有了前面的学习基础，还有两个学习与提高方向，一个是走 Java 或者 C# 的通用程序员之路，另外一个成为成为一名嵌入式程序开发人员。当然，建议你继续阅读更多的 C/C++ 著作，不断提升自己的“内功”，这对读者学习任何新的工具或者开发语言，都将会带来很大的帮助。

对于通用程序员之路，一般有三种选择，一个是以 Visual C++ 作为主要的开发工具，这在开发 Windows 相关系统类应用中比较多见，有了前面的基础，需要适当的 Visual C++ 项目开发训练。推荐训练之前学习《深入浅出 MFC》这本书。其次，是走 Java 或者 C# 的开发之路，这方面的好书很多，有了本书的基础，读者可以先选择一本语法类，快速强化训练掌握语法，然后根据自己的情况选择合适的图书项目和技巧开发类图书。

对于嵌入式程序员就业分支，在学习本书的基础上，可以再开发一个实践项目，封闭开发 1~2 个月。每天至少 4 小时，总共需要至少 100 个小时。经过一个比较大的项目的操作过程，应该即可成为一个合格的嵌入式开发工程师了。

辅助学习材料

在图书封底左上方的二维码下载包中，我们精心放置了以下资料：

- 《C 语言百问百例》和《C++语言百问百例》两套电子书。
- 包含 14 讲视频的 C 语言高级教程。
- 包含 19 讲视频的嵌入式 LINUX 培训计划教程。
- 本书源代码。

局限性和副作用

虽然竭力回忆我们成长之初的一些代码编写经验，但非常遗憾，现在只记得刚开始的时候，编写一个程序，会有满屏的错误需要纠正。至于如何度过这一阶段，在这一阶段积累了哪些经验，现在几乎都不记得了。也就是说，早期时候的代码调试经验，本书基本不具备，因为现在编写代码一般都是一次性通过，很少有语法问题，有问题也是隐藏得比较深的运行漏洞。另外，每个人的学习过程差异也很大。我们的作者中，有的英语很好，一看错误提示，很轻松就能解决问题。有的则靠词典，一个一个地解决，C 语言编写了一学期，英语水平却提高了不少，这是一种积极的副作用。

其次，每一个专业的程序员，到一定的时候，都只专注于自己的问题领域。虽然我们也邀请了高校教师、其他领域程序员参与到本书的创作中来，但 C/C++ 应用面极广，难免有些见解有所偏颇，尽信书则不如无书，当你通过本书完成阶段性过渡后，需要放下本书，寻找新的帮助。

还有一种 C++ 程序设计学习观点，在面向对象时代，应该先建立面向对象的开发思想，直接学习 C++。假如工作需要进入嵌入式开发领域，再学习 C 语言。这种先建立面向对象，然后学习面向过程的思路，我们有一定程度的认同。一方面，我们都是先学习了 C 语言，然后再学习 C++，这方面经验比较丰富。另一方面，我们调研发现，绝大多数高校，依然是保持先学习面向过程的 C 语言，然后学习面向对象开发语言的教学过程，所以，本书按照大多数读者的学习模式设计。如果您恰好是先学习的 C++，我们建议您先从本书第 4 篇开始学起。

另外，本书的大多数作者没有在大学课堂教学的经验，有些讲解方法，可能没有考虑到读者当前的客观情况，如果你学习本书时有一些困惑，还请读者和我们联系。

多人智慧胜一人

对于 C/C++ 的学习，除图书之外，我们认为，不断的上机实践是尤为重要的。没有笨学生，只有懒学生。长期的坚持是非常重要的，这就看个人的毅力了。

一个比较好的建议是：如果你是在校的学生，建议你同时再找一个同学，和你一起学习 C 语言。根据经验，如果有两三个同样爱好的同学一起学习。可以互相竞争、互相促进。而且，有了问题大家互相交流，学起来特别快。

当然，更主要的是靠自己，只是有些时候，旁观者清，同学可能可以轻松地发现代码

中的问题所在，而我们却要调试半天。

如果没有同学或学长一起学习，在网上找一两个牛人请教也特别重要。为了促进本书读者的学习，我们特地在网上提供了一个读者之间交流的 QQ 群：16900070。同时，在 QQ 群的共享文件中也提供一些免费的学习材料，作为本书的补充。欢迎读者在学习过程中下载使用。

不积跬步，难行千里

在十多年的程序设计学习与实践过程中，有一个同学的经历让我特别感慨：“

大学时候有个同学，爱好编程。大四做毕业设计的时候，发现他水平比我们高一截。因为很多大程序编译和调试的经验，你不去干，是明显不会知道的。

有问题我一般都去问他，和他探讨，当然，水平太低的问题他也不感兴趣。

有一次我无聊说到这个开发提升过程，夸奖了他一下。他说，哪里，我只是把你们打游戏的时间，用来写代码了。

你想想啊，其实我平均一周去三次机房。一学期只有 20 周比较有空，两年也就 80 周，240 次上机。再说，我家条件也不好，也不能保证我有更多的上机费用。

两年前我发现学校机房的开放策略后，开始我只能去上机练习一下 DOS 命令，还有就是 C 语言书上的小案例。后来想开发一个压缩小工具，不断“折磨”自己，想代码，练习，上机去测试，查资料，代码越写越长，水平才慢慢提高的。

你以为我天生就明白 exe 文件的原理啊，我也是看了多少书，做了很多次试验才明白的。”

之所以感慨，因为他的水平提高经历没有什么特别的，就是坚持每周上机 3 次罢了。但万事贵在坚持，开始可能只能编写一些小程序，只能验证一下语法，但量变会产生质变，当你逐渐能快速地写出数千行代码的一个小项目，开发一个小工具。你就已经具备了专业程序员的潜质，你可以很轻松成为一个专业程序员了。

而这些不过是要求你按照一定的学习路线，一步一个台阶，坚持就是胜利。大多数时候，完美的学习计划和一时的热情，敌不过长期的坚持。

好了，现在方法有了，工具有了，学习材料有了。我们开始逐步来完成学习目标吧！祝你成功！

编者

2018 年 3 月

目 录

Contents

第 1 篇 夯实 C 语言核心基础

第 1 章 Hello, C 语言

1-1 闭上眼睛细数 C 语言的特点.....	2
C 语言精巧	2
支持结构化程序编写模型	2
变量和语句是 C 语言的细胞.....	3
数学运算和逻辑表达式是 C 语言的第一个难点.....	3
函数是 C 语言赐给我们最有力的武器.....	3
数组是我们第一次学习组织数据	4
指针是 C 语言的精华, 但它是魔鬼发明的.....	4
struct 是我们临摹世界的画板	4
掌握基本概念的组合才是成为 C 语言高手的必经之路.....	4
字符串是指针确认的	5
位运算是 C 语言底层开发的特色.....	5
预处理命令是程序开发的脚手架	5
文件是一个序列	5
1-2 从基本概念开始	6
1-3 掌握 C 语言核心, 逐步扩张.....	8
1-4 C++时代的 C 语言学习.....	11
1-5 C++: 更好的 C.....	12
新的注释模式	13
新的输入/输出方法	13
const 的新功能.....	13
变量的定义更加灵活	13
函数的重载	13
带默认参数的函数	13
变量的引用类型	13
新的内存分配函数 new 和 delete	14
内嵌 (inline) 函数	14
作用域运算符	14

第 2 章 建立起程序设计基本概念

2-1 软件与程序	15
-----------------	----

2-2	程序与可执行文件	16
2-3	程序与源程序	17
2-4	源程序与程序设计语言	18
2-5	程序设计的基本过程	19
2-6	文件和目录	19
2-7	理解编码上机调试和步骤	22
2-8	C/C++程序员成长经验	23
	第一阶段: C 语言的学习	23
	第二阶段: 课程设计和毕业设计	24
	第三阶段: 硬件开发和学习 C++	25
	第四阶段: 中型项目开发经验	25
	第五阶段: 大型项目开发经验	25
	总结个人学习成长的道路	25
	因材施教: 使用两套教材	26

第3章 在 Visual C++中验证 C 程序

3-1	选择 Visual C++的三个理由	27
3-2	调试程序的五个步骤	28
3-3	编写 Console 应用	32
3-4	跨过调试程序的最初障碍	35
	语法错误 (Syntax Error)	38
	语义错误 (Semantic Error)	38
	运行错误 (BUG)	39
3-5	Visual C++简易使用参考	40
	新建文档	40
	打开文档	41
	查看文档	42
	保存文档	44
	在打开的文档中搜索文本	45
	在已打开的文档中替换文本	45
	在磁盘文件中搜索文本	46
	常规表达式搜索	46
3-6	从 Turbo C 到 Borland C++	47
3-7	理解函数库和头文件	48

第4章 代码基本逻辑：顺序、选择和循环

4-1	流程图	51
4-2	顺序结构	52
	【示例 4-1】输入三角形的三边长, 求三角形面积	52

4-3 选择结构	52
4-3-1 if 语句	52
【示例 4-2】if 语句的基本用法 (输入数值, 如果大于 0, 输出“正数”)	53
【示例 4-3】利用 if-else 语句对数值进行正负的判断	54
4-3-2 switch 语句	54
【示例 4-4】利用 switch 判断用户输入的日期是星期几	55
【示例 4-5】判断输入的日期是星期几 (使用 break 语句)	56
【示例 4-6】利用 if 语句根据成绩范围给出等级	56
4-4 循环结构	57
4-4-1 利用 goto 语句和 if 语句构成循环	57
【示例 4-7】计算 $1+2+3+\cdots+100$ 的值	57
4-4-2 while 语句	58
【示例 4-8】利用 while 循环语句计算 $1+2+3+\cdots+100$ 的值	58
4-4-3 do-while 语句	58
【示例 4-9】利用 do-while 循环语句来计算 $1+2+3+\cdots+100$ 的值	59
4-4-4 for 语句	59
【示例 4-10】小高斯的 for 循环语句实现方法。	60
【示例 4-11】小高斯的 for 循环语句实现方法之二 (省略表达式 2 的应用举例)	60
【示例 4-12】小高斯的 for 循环语句实现方法之三 (省略表达式 3 的应用举例)	60
【示例 4-13】小高斯的 for 循环语句实现方法之四 (省略表达式 1 和表达式 3 的应用举例)	61
【示例 4-14】当需要无限循环时, for 循环语句表达式全部被省略的应用举例	61
【示例 4-15】小高斯的 for 循环实现方法之五 (for 语句中的逗号表达式)	61
【示例 4-16】读懂奇怪的 for 循环 (for 循环语句中的表达式 2 为特殊表达式时)	61
【示例 4-17】利用 while 循环求满足 $n^2-15n-36>0$ 的最小整数	62
4-4-5 break 语句	63
【示例 4-18】演示 break 语句的应用: 计算 $1+2+3+\cdots+100$ 的值	63
4-4-6 continue 语句	63
【示例 4-19】演示 continue 语句的应用: 计算 1 到 100 的偶数之和	63
4-5 本章综合实例	64
4-5-1 综合实例 1: 求某整数段区间中的素数, 并输出	64
4-5-2 综合实例 2: 根据用户输入的年份判断年份是否为闰年	66

第 5 章 加强对函数的理解

5-1 函数的引入	67
【示例 5-1】求从 1 加到 100 的值	67
【示例 5-2】求从 1 加到 n 的值	68
【示例 5-3】求从 1 加到 n 的值 (高斯算法)	69
5-2 函数的基本概念	69
5-3 库函数	70
5-4 函数的嵌套调用	71

【示例 5-4】在 max3 函数中实现嵌套调用	71
5-5 代码封装：模块化程序设计的起步	71
5-6 接口/实现思想的第一次体现	72
5-7 总结一下：函数帮你编写更好的代码	72
5-8 对比 Pascal 语言和 C 语言中函数使用的差异	74
第 6 章 总结：快速掌握 C 语言基础知识	

6-1 再次 Understand C	76
6-2 把基础知识串起来	77
6-3 综合实例 1：打印 Fibonacci 数列	78
6-4 综合实例 2：把 Fibonacci 数列中的素数找出来	79
6-5 综合案例 3：在 Fibonacci 数列中加入数组的应用	80
6-6 最后的综合性代码	81
6-7 成为 C 程序员你还需要知道的事情	82
6-8 三小时你可以学会 FORTRAN	85

第 2 篇 提升你的编程功力

第 7 章 逐步写出大程序

7-1 准备案例 1：闰年的判断	88
7-2 准备案例 2：这一天是星期几	89
7-3 更高要求：万年历的编写	91
7-4 再完善一下：打印某年日历	94
7-5 总结与思考：程序成长与模块化	96

第 8 章 自己动手编写小工具

8-1 DIR 命令今犹在	97
8-2 DOS 版 DIR 命令	98
8-3 Windows 版本的 DIR	100
8-4 工具编写总结和练习	103
8-5 继续学习和提高	104

第 9 章 学会组织自己的代码

9-1 解决问题的基本方法和 C 程序的基本结构	105
9-2 用 Project 管理自己的代码	108
9-3 为自己的代码扩充功能	110
9-4 养成一些编码的好习惯	111

第 10 章 读源代码, 逐步体会算法的力量

10-1 读函数源代码, 学技巧	116
10-2 体会算法的力量	118
【示例 10-1】判断一个自然数 n 是否是素数	119
10-3 算法的三重境界	121
【示例 10-2】写一个函数计算当参数为 n (n 很大) 时的值 $1-2+3-4+5-6+7-\cdots+n$	121
【示例 10-3】在一个很小的内存空间中, 将北京某电话局的 8 位电话号码排序。号码在 10000 个以内	122
10-4 那些 C 语言必须调试的陷阱	123

第 3 篇 积累专业程序员的开发经验**第 11 章 烦琐但很重要的变量和类型**

11-1 计算机是如何存储变量的	126
11-2 确保每个对象在使用前已被初始化	131
11-3 局部变量和全局变量的差别	133
11-3-1 首先理解函数中的局部变量	134
11-3-2 其次理解函数外的全局变量	135
11-3-3 掌握变量在内存中的存储方式	136
11-4 掌握变量定义的位置与时机	138
11-5 引用难道只是别人的替身	140

第 12 章 理解数组和指针

12-1 理解指针的本质	143
12-1-1 指针变量和变量的指针	143
12-1-2 空指针和 void 指针	145
12-1-3 const 修饰指针	146
12-2 论数组和指针的等价性	147
12-3 再论数组和指针的差异性	149
12-3-1 字符串数组与字符指针的区别	149
12-3-2 当编译器强行把指针和数组混合	151
12-4 充满疑惑的数组指针和指针数组	152

第 13 章 常见更要谨慎的字符和字符串

13-1 关于字符编码的讨论	155
13-2 请牢记字符串结束标志为 ‘\0’	157
13-3 请务必小心使用 memcpy() 系列函数	159
【示例 13-1】POD 对象的复制	160