

前言

程序设计课程在计算机科学教育中的重要性是无庸置疑的。人们一直在思考第一门程序设计课应该教授什么，其教材应该是怎样的？在用 C 作为第一门程序设计课教学语言的过程中，我编写了这个教材，是对于自己教学经验的总结。

教学过程中学生们提出的问题使我看到，教材（或适合自学者的书籍）中需要对可能的问题做出合理而充分的解释，无论是关于 C 语言本身，还是关于程序设计过程，使读者在阅读时能了解问题实质，而不是在模糊中去设法感悟。这对于帮助初学者尽快进入程序领域非常重要，也使教师有更大自由度，灵活选择实例和教学方式。由于程序设计的性质，这里的实例不应采用“提出问题，给出解答，再加点解释”的简单三步形式，而应着重帮助读者认识程序设计活动的实质，理解从问题到程序的思考过程。本书在这些方面做了些努力。

本书的一个基本想法是希望反映程序设计的两个重要特征，科学性和工程性。科学性指程序构造过程应有充分科学依据：分解弄清问题各部分的意义和互相联系，需要人对程序实现过程有科学认识。有关研究的发展形成了程序的理论。入门书不可能去讨论这些理论成果，但也必须反映其精神实质，使初学者从开始就接触程序的一些本质性问题。这无论对于提高读者对计算机科学技术认识水平，还是对他们的进一步深入学习，都是非常重要的。

随着计算机的日益普及，不少人在进大学前就有了计算机经验，有的已将计算机玩得很熟，甚至写过许多程序。但也应看到另一面：在学习大学课程前，学生中对程序及程序设计已建立起正确认识的人很少。即使那些很熟悉计算机的学生，也有许多观点需要建立，或有一些错误观点需要纠正。这也是本课程中应该特别强调科学性的原因。

我们常发现学生有些对计算机实质的不正确认识。例如有些学生在头脑深处把计算机看成很不驯服的东西，觉得程序设计就是设计一些计谋，“骗”计算机干活。有些学生把程序设计看成玩计算机游戏，到处是陷阱和妖魔，需要在试探和失败中找一条绕过障碍的路。

日益完善的程序开发环境也可能成为学习的障碍。很好的环境造就出一类学生：遇到编程工作时不细致分析和设计，粗一想就动手，很快写出一堆代码，随后进入运行、追踪、打补丁改错的三步循环。简单程序可能这样做出来了，但质量很差，很简单问题常常写出一大篇，自己也说不清解决问题的过程。即使很熟悉程序的人，不花些工夫也难判断这类程序的对错，错误在哪里。当问题更复杂时，这些同学就遇到了困难：程序难以通过测试，弄不清错误出在那儿，该怎样修改。这种程序像建在沙滩上的大楼，无论如何修补都不行，只能推倒重来。这些情况的提示是：从简单程序开始就应该注意程序设计的正确途径。

这些都说明了强调程序设计过程中科学性的必要。本书从开始就强调问题的分析和分解，反复讨论函数抽象方法。后面不断有进一步论述，实例中也特别注意这方面的问题。一些地方还通过实例介绍了一些更深入的理论问题，如通过对程序运行时间的测试介绍计算过程的基本性质（复杂性）；通过分析循环过程是否完成了所需工作，介绍“循环不变关系”的概念和意义等。当然，这些只是希望读者了解有关情况，作为思考程序问题的线索。

这里说程序设计的工程性，并不是指规模大，参加人员多等，而是从另一个角度看问题。工程设计中需要分析和理解问题，寻找可能解决方案，对各种方案做出评价和选择。对所做选择应有清醒认识（优点和缺点，是否对问题的某些方面有所偏向或不足等），进而确定具体实施方案。这些问题在程序设计中都有直接反映。如果学习程序设计的目的是希望提高能力、真正理解程序设计过程的话，这些问题都需要特别强调。

本书始终强调一种观点：程序问题并没有需要记住的标准答案。由于分析问题时的不同考虑，设计过程中的不同决定或选择，人们对同一问题常会得到许多合理而正确的不同程序。

这些程序常各有长短，可能各有侧重点，也可能反映了对问题的不同认识。应该特别注意思考 and 解决问题的方法，包括如何分析问题，逐步把问题弄得更清楚明确；如何寻找可能求解途径，把复杂问题分解为相对简单的步骤；如何在可用的语言结构中做出选择等。这里的每一步都可能产生分支，应该认清各种选择的后果，无论是收获还是损失。

工程中通常没有完美的选择，更多的是权衡和折衷，这些对程序设计也是本质性的。本书对许多实例给出了较详细的分析过程，常常对一个问题给出多个解答，有时还指出了其他可能性：还可能如何看问题，还可能如何做等等。这里还常常给读者提出一些问题，希望读者发挥自己的思维能力和主观能动性。各章节后面的练习也试图反映这些看法。

我们希望读者能建立一种认识：程序设计问题并没有什么“标准答案”，要追求的是比较好的“正确”答案，各种书籍中给出的答案（包括本书）不过是作者对问题理解和分析的反映，还有很多可能选择。进一步说，我们还希望读者养成这样的习惯：在看别人的程序时，应分析其中隐含着作者的哪些考虑和选择，哪些是合理的有价值的（或不合理没价值的）？还可能有什么选择？沿其他选择走下去可以得到什么（或失去什么）等等。这种思考习惯必将使人受益匪浅。当然，这并不是说书中的示例不重要。恰恰相反，因为程序设计中有许多选择可能，书中更应当给出合理的好的选择供读者参考。入门书籍还应当说明理由，如果可能，还应当指出采取这些选择带来的问题（缺点，限制等等）。

正如本书的书名所言，程序设计是“从问题到程序”的思考和工作的过程，这里既要求掌握严格的科学方法，又要求谨慎灵活的工程能力。要很好地完成程序设计工作，编程者需要充分发挥聪明才智，还要有细致认真、一丝不苟的工作态度。即使将来不从事程序工作，这一课程中的锻炼也可能很重要，尤其对于理科学生，可能弥补其工程方面训练的不足。

前些年的程序设计课常用 PASCAL、FORTRAN 或 BASIC 等语言，目前已转向 C 或其他类 C 语言（如 C++）。从作为入门课教学语言的角度看，没有一种语言具有无可比拟的天然优势，选择时都应考虑其有利方面，也需要克服这种选择带来的不利因素。

选择 C 作为教学语言的主要理由有：C 是目前实际使用最广的语言之一，它包含了需要理解的基本机理和主要机制，能满足讨论程序与程序设计问题的需要。学生可以用它完成程序练习，得到有关知识积累和能力锻炼，还能掌握一种实用的工具。另一方面，程序设计是计算机领域的基础课，学了 C 语言后，后续课程比较容易安排。C 语言适合（也正被）作为计算机领域许多课程的教学语言。第三，C 是一种很灵活的语言，用它写程序常需要了解一些细节，这是人们对用 C 作为基础课语言的主要疑虑。问题也有另一面，通过对 C 语言程序设计的较好理解，读者也可能得到更深入的认识。这个问题实际上对课程组织，教材和教师提出了较高要求。C 程序可以在较低层次上做，也可以在较高层次上做，学生可能从中更多了解程序设计过程中的各种问题。此外，许多语言从 C 借鉴了想法和形式，有些是 C 的扩充和发展，C 语言知识对于进一步了解其他语言，包括未来的新语言都有价值。

在撰写本书时我心中有几个努力追求的目标，列在这里供读者和同行参考：

1. 假定读者（基本上）没有程序设计经验，希望本书适合作为入门课教材。
2. 以程序设计为基本线索，同时对 C 语言做深入介绍，强调如何认识程序、写程序和用 C 写程序。通过实例讨论问题分析和分解，设计求解过程，找出主要步骤，确定函数抽象，找出循环，选择语言结构，最后写出程序的过程。不少实例给出了在不同考虑下可能形成的多种解法，以帮助读者理解程序设计的真谛。
3. 强调好的程序设计风格，通过函数抽象建立起清晰结构的重要性。尽可能早地引进函数概念，从库函数、简单函数定义等等。强调程序的结构性、可读性、易修改性。书中还根据进展和遇到的问题分析了一些不良程序设计习惯及其危害。

4. 注意强调“好的”C程序设计。历史使C成为一个不太严格的语言，如不注意，用C写的程序常会隐含不易发现的错误*。ANSI C标准倡导了一套写“好的”C程序写法。本书力图坚持这种正确途径，讨论了如何写出更可靠、不易包藏隐含错误的C程序的各方面问题，通过实例说明了应该如何写和不应该如何写。在此基础上也介绍了一些实用的C程序设计技术。本书希望强调的是如何写正确、清晰、简洁、高效的C程序。
5. 详细介绍和解释了C的各种结构和机制，因为其中不少问题反应了相关领域的普遍性知识和情况。这里还尽可能提供一些背景理由，以帮助读者理解问题实质。对语言和程序设计的一般性问题及计算机科学中的一般问题也有适度介绍。

本书包括如下各章和若干附录：

第一章，程序设计与C语言。介绍程序与程序语言的概念，C语言的发展及其特点。用一个小例子介绍C程序形式，其加工和执行。最后介绍程序设计与开发过程。

第二章，数据对象与计算。讨论程序语言的许多最基本概念，包括：字符集、标识符和关键字，数据与类型，数据表示，运算符、表达式与计算过程，数学函数库的使用等。

第三章，变量、函数和控制结构。讨论程序设计的一些基本问题，包括语句与复合结构，变量及其使用，简单函数定义，逻辑条件的描述与使用等。最后介绍了几种基本控制结构。

第四章，基本程序设计技术。首先分析循环程序设计的基本问题，通过一系列程序实例分析了循环的构造过程。此后介绍了C语言其他控制结构及其使用。

第五章，C语言程序结构。讨论C语言许多具有一般性的重要问题，主要是C程序结构，函数概念及有关的问题，预处理命令和预处理过程，递归的概念等。

第六章，数据对象的顺序组合：数组。介绍数组概念、定义和使用，数组与函数的关系，二维和多维数组等。

第七章，指针。指针概念和指针变量的使用，C语言中指针与数组的关系，多维数组作为参数的通用函数。而后讨论了动态存储管理，类型定义，指向函数的指针等一系列问题。

第八章，输入输出与文件。讨论了文件概念，与输入输出有关的各种问题，标准库的输入输出功能，以及输入输出的程序设计问题。

第九章，结构及其他数据机制。介绍结构(struct)、联合(union)、枚举(enum)等数据定义机制的意义及在程序中的使用。随后简单介绍了链接结构的概念。

第十章，程序开发。讨论程序设计和开发中的许多一般性问题和技巧。包括C程序的分块开发问题等。

第十一章，标准库函数。介绍标准库提供的各方面功能及其相关知识。

最后有几个附录和一个比较详细的索引。

本书以标准C为背景，有关内容不依赖于具体C系统。读者可用任何符合ANSI C标准的C系统作为编程环境，如国内常见的微机上的各种C系统，工作站或其他计算机上的C系统。虽然软件厂商不断推出新版本的程序开发环境，但从学习基本程序设计的角度看，新开发环境比早先版本并没有实质性改善，通常是更复杂、占用更多计算机资源，初学者入门更困难。本书的实例程序按ANSI C标准书写，习题也不涉及任何具体系统环境，因此建议读者采用比较简单基本的系统作为学习工具，例如国内使用较多的TURBO系列C系统，一些公开的免费C语言系统等。本书最后附了对TURBO C系统的简单使用说明。

我特别感谢北京大学数学学院和北京大学理科试验班参加课程的同学和参加课程辅

*任何语言都有弱点。有句名言说：“再好的语言也不能阻止人写出坏程序”。这不是说语言不重要，而是说任何语言都有合理使用，写好程序的问题。C在这方面的问题突出一点，读者应特别注意。

导的研究生们，是他们的思考和问题给了我许多启示，使我更深入地理解了许多问题。我也要感谢我的家人与同事在这些年工作中给我的支持。

虽然本书包含了我几年的工作和思考，但书中难免有大的或小的错误，这些都由我个人负责。希望读者能把发现的问题告诉我，也希望同行们对本书提出宝贵意见。

裘宗燕

北京大学数学学院信息科学系，1999年1月

2003年修订