重点大学计算机专业系列教材

计算机语言与程序设计

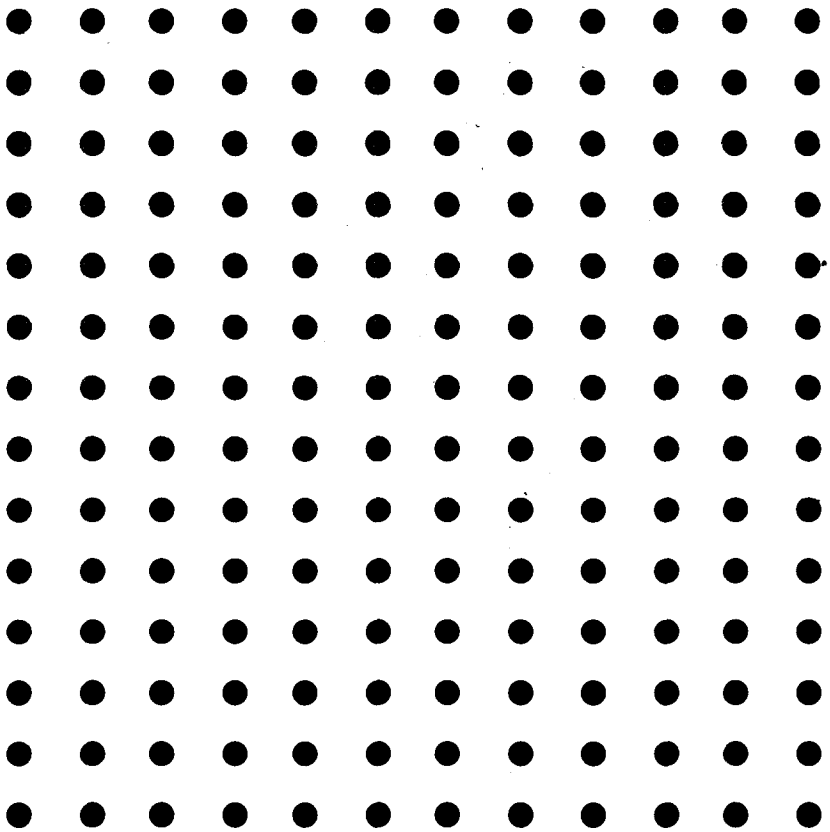
谌卫军 编著

清华大学出版社

重点大学计算机专业系列教材

计算机语言与程序设计

谌卫军 编著



清华大学出版社
北京

内 容 简 介

本书是清华大学信息学院本科生的教材，主要介绍 C 程序设计语言的基本知识，包括数据类型、选择语句、循环语句、数组、函数、指针、结构体和文件等；同时还介绍程序设计的基本方法、技术和理论。

本书中结合了作者多年讲授相关课程的教学经验以及长期的工程实践经验，也融入了最新的教学改革理念，即从知识型教学向技能型、实用型教学转变。在介绍 C 语言语法知识的同时，更注重培养学生分析问题、解决问题和程序设计的能力。本书内容详细、语言风趣，采用实例教学。书中附有大量的习题，便于自学。

本书适合作为高等院校计算机、自动化和电子等专业本科生的程序设计教材，也可作为正在学习程序设计的广大科技人员、软件工程师的参考教材。

图书在版编目（CIP）数据

计算机语言与程序设计 / 谌卫军编著. —北京：清华大学出版社，2007.9
（重点大学计算机专业系列教材）
ISBN 978-7-302-15434-1

Ⅰ. 计…　Ⅱ. 谌…　Ⅲ. C 语言-程序设计　Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2007）第 086865 号

责任编辑：丁　岭　赵晓宁
责任校对：李建庄
责任印制：王秀菊
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　**邮　　编**：100084
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175　**邮购热线**：010-62786544
投稿咨询：010-62772015　**客户服务**：010-62776969
印 刷 者：北京市世界知识印刷厂
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：20.25　**字　数**：464 千字
版　　次：2007 年 9 月第 1 版　　**印　次**：2007 年 9 月第1 次印刷
印　　数：1～4000
定　　价：27.00 元

INTRODUCTION

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大，社会对计算机专业人才的需求不仅体现在数量的增加上，而且体现在质量要求的提高上，培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前，我国共有16个国家重点学科、20个博士点一级学科、28个博士点二级学科集中在教育部部属重点大学，这些高校在计算机教学和科研方面具有一定优势，并且大多以国际著名大学计算机教育为参照系，具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系，形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础，其中专业教材的使用和建设则是这种氛围的重要组成部分，一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势，特别是专业教材建设上的优势，同时配合各重点大学的计算机学科建设和专业课程教学需要，在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下，清华大学出版社规划并出版本系列教材。本系列教材的建设旨在“汇聚学科精英、引领学科建设、培育专业英才”，同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

1．面向学科发展的前沿，适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础，反映基本理论和原理的综合应用，重视实践和应用环节。

2．反映教学需要，促进教学发展。教材要能适应多样化的教学需要，正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、

素质协调发展创造条件。

3．实施精品战略，突出重点，保证质量。规划教材建设的重点依然是专业基础课和专业主干课；特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现重点大学计算机专业教学内容和课程体系改革成果的教材。

4．主张一纲多本，合理配套。专业基础课和专业主干课教材要配套，同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系；基本教材与辅助教材以及教学参考书的关系；文字教材与软件教材的关系，实现教材系列资源配套。

5．依靠专家，择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主编。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

FOREWORD

前言

《计算机语言与程序设计》是清华大学信息学院各院系为大一新生开设的一门公共基础课。课程以程序设计的基本原则为线索，以 C 语言为工具，介绍程序设计的基本方法与 C 语言的基本语句，目标是使学生具备用 C 语言进行程序设计的能力。为了配合课程的讲授，便于学生的学习，我们编写了本教材。

C 语言是一种非常重要的编程语言。一方面，由于它的高效、实用和可移植性好等特点，使它在工业界得到了广泛的应用，许多实用的软件（如操作系统、数据库管理系统等）都是用 C 语言编写的；另一方面，由于它的语法简洁、紧凑，容易入门，使它取代了 Basic 和 Pascal 等语言，成为了高等院校甚至某些中小学计算机教育的入门语言。不过，需要指出的是，虽然 C 语言的语法并不复杂，但要想真正学好它，却并非易事。在多年的教学实践中，我们深刻地体会到了这一点。由于 C 语言程序设计是学生第一次接触到编程，因此缺乏相关的基础和经验，更重要的是，缺乏正确的学习方法，以为吃透大纲、熟读教材、多做习题就能把编程学好。具体表现为：在学习时偏重于语法和概念的钻研，死抠一些烦琐的语法细节，喜欢对一些古怪、偏僻的问题刨根问底，如计算“(i++) + (i++)”、“a += a -= a *= a”等，以为这些才是高水平的表现，其实不然。就像孔乙己去研究“茴”字的不同写法，人们只会说他迂腐。另外，在课后练习时，采用的也是应试教育的方法，喜欢去做一些选择、填空等书面的传统题型。这样的结果，就是容易使自己陷入眼高手低、纸上谈兵的误区。事实上，我们在教学中最常碰到的一个问题就是：学生对 C 语言的语法都非常熟悉，但是一旦要让他们用编程来解决一个实际的问题就不行了。另外，学生在毕业找工作时，不少单位抱怨，大学培养出来的学生都说自己学过 C 语言，但无论是在解决问题的能力上，还是在编程的规范性上，离公司的要求都相距甚远，还得重新培训。

那么如何来解决这个问题呢？我们认为，关键在于教师的正确引导。

对于C语言程序设计这样一门实践性强、内容广、学生起点低的课程来说，不能采用传统的教学模式、学习模式和训练模式，而必须另辟蹊径。具体来说：

1. 在教学理念上，应该根据社会经济发展对人才的需求来研究课程的建设和学生的培养。要从学历教育向素质教育转变，从知识型教学向技能型、实用型教学转变。本课程的目标不仅仅是让学生掌握程序设计的基本原理和方法，更重要的是提高他们的能力，包括自学能力、动手实践能力和创新能力。首先，程序设计类课程的一个特点就是语法和技术细节很多，这些内容比较枯燥、单调，不适合在课堂讲授。但如果学生在上机实践时碰到这些问题，能够很快找到解决方案，并且印象深刻，因此自学能力是非常重要的。其次，动手实践能力可以说是本课程的灵魂。在教学中我们经常发现，在拿到一个问题后，有的学生觉得自己会做，而且思路也很清楚，说起来也有道理，但是一旦要让其上机编写出相应的程序，并且正确地运行，就卡壳了。一道简单的题目，被卡上几个小时是常有的事，被卡上几天的情形也不少见。而出现这些现象的根本原因就是缺乏有效的、足量的实践训练。最后，程序设计类课程为学生提供了一个自由发挥的平台。在这个平台上，没有权威，没有定理，对任何一个程序设计问题都存在不同的解法，“条条道路通罗马”。学生可以充分发挥自己的想象力，充分运用所掌握的基础知识，设计出各种与众不同的解决方案，从而提高自己的创新能力。

2. 在教学内容上，除了C语言程序设计的基础内容，如C语言的语法、语句和数据类型等，还应该引入一些实用、创新型的内容。例如，在函数这一章，通常的教材只是介绍函数的使用方法，而本书则引入了函数调用的实现过程，以及栈和堆的基本概念，从而使学生不仅知其然，而且知其所以然。此外，还有编程规范、代码的调试与测试等，对于一名专业的程序员来说，这些都是最起码的要求，几乎每天都要用到。而在大学的课堂上却容易被忽略。

3. 在教学方法上，我们始终坚持“案例教学”和“模式教学”的思想。对于程序设计类课程，案例教学是一种极其重要的教学手段，在每一个案例中都蕴藏着程序设计的理论和方法。因此，通过案例教学，可以实现具体与抽象、理论与实践的有机结合，从而培养学生分析问题、解决问题的能力。在本书的每一章节，我们在介绍理论知识的同时，精选了大量具有代表性和实用性的新颖案例，从而将知识的讲解融入到案例的学习之中。对于每一个编程实例，我们不是简单地把源代码列在那里，然后解释一番，而是着重于问题的分析和解题的思路，讲清算法的来龙去脉，并将一些常用的算法思路抽象为若干个“编程模式”，好比是武术里面的“套路”。让学生知道，对于什么样的问题，可以使用哪一种模式，或是使用哪几种模式的组合来解决它，从而切实提高学生的动手能力。另外，读了这么多年的书，我们完全能够理解，读书是一件比较辛苦的事情，时间长了难免疲劳。因此在本书中，我们穿插了不少生动有趣的例子。例如，在讲授循环语句时，读者将看到西绪福斯的故事，还将学到价格竞猜的秘诀；在学习递归算法时，读者将听到熟悉的“从前有座山”的故事。

4. 在实验环节上，我们强调一个原则，即上机实践是学好本门课程的唯一途径。推而广之，要想学好程序设计类课程，都要坚持“机考”、“机练”的原则。为此，在每

一章的后面，我们提供了大量的精心挑选的编程习题。这些习题是本课程 4 年以来的积累与总结，具有较好的代表性。希望读者能善于利用，最好每一道题都做，并上机调试通过。

本课程已经实施了 4 年多的时间，在教学实践过程中，我们根据各方面的反馈情况，不断地进行更新和完善，同时，这门课程也得到了专家和学生的认可，取得了较好的成绩，目前已进入了较为成熟和稳定的阶段。2003 年秋季，本课程获清华大学软件学院 2003 年度教学优秀奖。2004 年秋季，本课程在教学评估中，在软件学院的本科生课程中排名第一。2005 年秋季，本课程在教学评估中，再次蝉联第一，并在清华 30～100 人课堂规模的分组中，获得全校第 8 名。2007 年 1 月，本课程参加清华大学第二届青年教师教学基本功比赛，我们的教学理念获得了各位评审专家的认可，获一等奖。本书就是对这 4 年来我们工作的一个总结，希望与各位读者分享。

在本书的写作过程中，得到了许多人的关心和帮助，在此一并表示感谢。其中，清华大学计算机系的吴文虎教授是本人教学工作的启蒙老师，通过观摩他的课堂教学，使我受益匪浅。谭浩强教授的《C 程序设计》是本人学习 C 语言的启蒙之作。Stanford 大学的 Khomenko 教授和 Washington 大学的 Dickey 教授的讲义也给了我很大的启发。此外，感谢清华大学出版社的丁岭女士，没有她的辛勤工作，也就没有本书的问世。

最后，我要特别感谢我的父母和家人。你们的关心、理解和支持，永远是激励我不断前进的动力。尤其是我的女儿谌玥颖，你是上天赐给爸爸的礼物，爸爸希望用这本书来作为你的生日礼物。

谌卫军

2007 年 3 月 23 日于清华园

CONTENTS

目录

CHAPTER 1

程序设计概述 第1章

1.1 计算机与程序

1.1.1 功能强大的计算机

计算机是20世纪人类最伟大的发明之一，它的发展历史可以追溯到中国古代的算盘。算盘是一种辅助计算工具，人们在进行算术运算的时候，无需复杂的心算，只要使用一些固定的口诀来拨弄几下算珠，就可以把答案算出来。1642年，法国物理学家帕斯卡利用机械齿轮原理，发明了第一部能计算加减法的计算机。1671年，德国数学家莱布尼兹发明了一种能作四则运算的手摇式计算机，这些工作都是早期的机械式计算机的代表。20世纪初，随着电子管的出现，计算机有了新的发展。1946年，由于第二次大战的军事需要，美国宾夕法尼亚大学和有关单位研制成功了第一台真正意义上的电子计算机——电子数字积分仪与计算机（Electronic Numerical Integrator and Computer，ENIAC）。几十年过去了，计算机取得了迅猛的发展，其使用的元件也经历了四代的变化：第一代的电子管、第二代的晶体管、第三代的集成电路和第四代的大规模集成电路。如今，电子计算机的功能已不仅仅是计算，它已渗入了人类的活动领域，成为人们工作和生活中必不可少的工具。在我们的周围，有着各种各样的计算机，如笔记本计算机、台式机、个人数字助理（PDA）、小型机和大型机等。

现代计算机的功能非常强大，能够为人类做许许多多的事情。比如说，在中国古代，一些书香门第或官宦家庭，为了培养一个人的人文素养，通常都要求他掌握4项基本的技能，即琴、棋、书、画。对于这4项技能，如果能做到样样精通，那么就是一个高雅的人，是一名文人雅士。当然，对于现代人来说，在“分数压倒一切”的背景下，不要说精通这4项技能，只要会

其中的一项就很不容易了。但是计算机就能够做到样样精通。首先是弹琴，这对于计算机来说是小菜一碟，计算机合成的音乐很早以前就有了。其次是下棋，这更是计算机的强项，计算机不仅能够下棋，而且下得非常好。1997 年的 5 月，由 IBM 公司开发出来的“深蓝”计算机，就战胜了当时的国际象棋世界冠军卡斯帕罗夫。卡斯帕罗夫是俄国人，曾被认为是有史以来最厉害的棋手之一，但他却被计算机给打败了。当然，古人所说的棋指的是围棋，而围棋比象棋要复杂得多，所以在这个方面，计算机还很难有大的突破。第三种技能是书法，这对于计算机来说，是再简单不过的了，什么样的字体它都能打印出来，如宋体、楷体、隶书等。第四种技能是画画，这也不是什么难事，在计算机的控制下，能够在布匹上刺绣，甚至还能编织任意图案的毛线衣！

除了琴棋书画，计算机还能做到听说读写。所谓的“听”，指的是语音识别技术，也就是说，人对着计算机说话，然后计算机就会把这些语音信号转换成相应的文字。比如说，读者需要把一篇稿子录入到计算机当中，但是又不想通过键盘输入，因为通过键盘输入得太慢、太辛苦了，这时就可以使用语音识别软件。你只要对着话筒把这篇稿子念一遍，它就被录入计算机了，非常的方便。当然，这只是一种理想状态，而实际的语音识别软件目前还做不到这么完美，它们的识别正确率还达不到百分之百。但即便如此，这项技术在文字录入、身份认证等领域还是得到了广泛的应用。所谓的“说”，指的是文语转换技术（Text-To-Speech，TTS）。它和语音识别正好相反，一个是把声音变成文字，另一个是把文字变成声音。比如说，我们上网去看今天的新闻，但是又不想用眼睛去看，因为看得比较累，这时就可以使用 TTS 技术，让计算机把新闻念给你听。再比如，我们都知道英国著名的理论物理学家霍金，他是一个全身瘫痪的人，除了大脑以外，身体的其他部分都是处于肌肉萎缩状态，连说话都不行。那么他是如何与别人交流的呢？就是通过 TTS 技术。当他想要说话的时候，就用手指尖去操作一台计算机，输入他想要说的话，然后，计算机就采用 TTS 技术，把这句话转换成语音播放出来。TTS 系统的关键指标是自然度，一般来说，计算机合成出来的声音不是很好听，语调比较生硬。就像电影《星球大战》中的机器人 C-3PO，说起话来有点怪声怪气的。当然，现有的 TTS 技术已经取得了长足的进展，已经能够合成出比较好听、比较自然的声音。所谓的“读”，指的是自然语言理解，也就是说，对于一段自然语言文字（如中文、英文等），计算机能够看懂它的意思是什么。自然语言理解有很多的应用，如机器翻译，能够把一种语言的文字翻译成另一种语言；再比如因特网上的基于内容的智能搜索。我们可以向计算机提出各种问题，如“北京有什么好玩的地方？”，计算机能够看懂这句话的意思，然后经过搜索，就会把北京的一些旅游景点列出来，如故宫、颐和园、圆明园及长城等。所谓的“写”，就是说，计算机能够自动写作文。这其实也不是什么太难的事情，一般的做法是去搜集很多写得比较好的句子，然后把它们拼接在一起即可。曾经有人对某个国际学术会议的组织者表示不满，认为他们盲目扩大会议规模，而忽视了论文质量的审查。于是他就开了一个玩笑，把一篇计算机自动生成的“论文”提交给该会议，后来这篇论文居然被录用了[①]。

① 以上所说的这些技术，实际上都是计算机科学领域当中的一些研究课题。读者如果感兴趣，将来可以从事相关方面的研究。

那么，计算机为什么能够做这么多的事情呢？是不是它的内部结构非常复杂，比人的大脑还要复杂？答案是否定的。实际上，计算机的工作原理非常简单。从本质上来说，计算机是一种用来处理数据的通用机器。它所能够做的事情只有一件——计算。因此，从某种意义上来说，可以把计算机看成是一个超级计算器。

图 1.1 是计算机硬件的体系结构图。一般来说，一台计算机主要由三个部件组成，中央处理器（Central Processing Unit，CPU）、内存（memory）和各种输入输出（Input/Output，I/O）设备，如显示器、键盘、磁盘和鼠标等。其中，CPU 是计算机的“心脏”，所有的计算都是在 CPU 上进行的；内存是计算机的“大脑”，当计算机在运行时，所有的信息都存储在内存当中，包括指令和数据；而 I/O 设备是计算机的“手脚”和“五官”，对于用户来说，正是通过这些 I/O 设备来与计算机打交道。那么计算机是如何来工作的呢？很简单，就是从内存当中，取出一条指令，放在 CPU 上去运行。这条指令可能是一条算术运算指令，对两个数据进行加、减、乘、除；也可能是一条内存访问指令，去内存存储或读写数据；也可能是一条数据比较指令或控制指令。当这条指令执行完以后，系统又从内存当中取出下一条指令，放在 CPU 上去运行。就这样一条指令接一条指令地运行，直到程序运行结束。这就是计算机工作的整个过程。那么指令和数据是怎么样从内存跑到 CPU 上去的呢？它们是坐“公共汽车”（bus）去的。当然，这只是一个玩笑。实际上这里的英文单词 Bus，在中文里一般翻译为总线，指令和数据就是通过总线进入到 CPU 的。

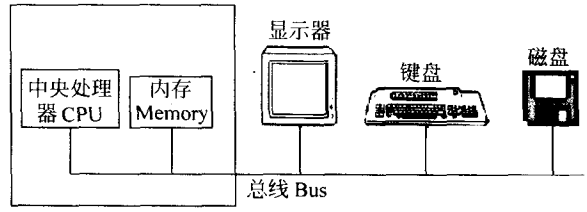


图 1.1　计算机体系结构图

通过刚才的介绍可以知道，计算机的工作原理其实是非常简单的，它就是在那里不断地执行指令，不断地计算。那么，为什么这么简单的工作原理，却能够完成那么多不可思议的任务呢？原因主要有两个，首先，计算机的运算速度非常快。比如说，如果让我们去心算一下 53×7 等于多少，那么我们可能需要 1 秒钟左右的时间才能给出正确的答案，但是对于奔腾III系列的 CPU 来说，它只需要十亿分之一秒就够了。“深蓝”之所以能够战胜卡斯帕罗夫，就是因为它算得实在是太快了，每秒钟能够算 2 亿步。这样，不管你采用什么招法，它只要进行遍历式搜索，把各种可能的走法都算一遍，然后从中选择一个最佳的走法即可。当然，在围棋方面计算机还暂时没有太大的突破，原因很简单：围棋的棋盘有 19×19 个格子，361 个点，每个点有 3 种可能，即黑棋、白棋或空白，这样，棋盘的每一个状态就有 3^{361} 种可能，这是一个天文数字。除了运算速度快，计算机的第二个优点在于它的精度非常高，它可以不知疲倦地在那里计算，而且不会出错。

例如，根据一项统计数据，计算机在访问磁盘的时候，每隔 10 亿个数据位才可能会发生一个错误。所以说，计算机的工作原理虽然很简单，但由于它速度快、精度高，所以能够实现很多不可思议的功能。

虽然计算机的功能非常强大，但需要指出的是，计算机并不能直接帮助人们去解决问题。因为人们在描述这些问题的时候，采用的都是人类的自然语言的形式，而不是机器指令的形式，这样的话，计算机就听不懂人们所说的话，更不知道该去做什么。比如说，假设我们想要知道 1 加 1 等于多少，那么我们不能拿起话筒，然后对一台计算机说："计算机，请你告诉我，1 加 1 等于多少？"这样做是不行的，计算机根本就不会搭理你，因为它听不懂你说的话，不知道你想要的是什么。那么我们要怎么做，才能够让计算机来帮助我们解决这个问题呢？答案就是计算机程序！

1.1.2 计算机程序

所谓的计算机程序（computer program），就是计算机能够识别、执行的一组指令。如前所述，人们正是通过编写程序（programming）来让计算机帮助我们解决各种各样的问题。这个过程一般可以分为以下的 4 个步骤。

（1）需求分析。当我们拿到一个问题以后，首先要对它进行分析，弄清楚我们的核心任务是什么，输入是什么，输出是什么等。比如说，假设我们要编写一个程序，实现从华氏温度到摄氏温度的转换。显然，对于这个问题来说，输入是一个华氏温度，输出是相应的摄氏温度，而我们的核心任务就是如何来实现这种转换。

（2）算法（algorithm）设计。对于给定的问题，采用分而治之的策略，把它进一步分解为若干个子问题，然后对每个子问题逐一进行求解，并且用精确而抽象的语言来描述整个的求解过程。算法设计一般是在纸上完成的，最后得到的结果通常是流程图或伪代码的形式。

例如，对于上述的温度转换问题，我们可以设计出如下的算法：

① 从用户那里输入一个华氏温度 F；

② 利用公式 $C=\frac{5}{9}(F-32)$，计算出相应的摄氏温度 C；

③ 把计算出来的结果显示给用户看。

（3）编码实现。在计算机上，使用某种程序设计语言，把算法转换成相应的程序，然后交给计算机去执行。如前所述，我们只能使用计算机能够看懂的语言来跟它交流，而不能用人类的自然语言来命令它。

（4）测试与调试。最后一个步骤是测试与调试程序。我们在编写程序的时候，由于疏忽，经常会犯一些错误，如少写了一个字符、多写了一个字符或拼写错误等，但是计算机是非常严格的，或者说是非常苛刻的，它不允许有任何的错误存在，哪怕是再小的错误，它也会给你指出来。所以在编完了程序以后，我们通常还要进行测试和调试，以确保程序能够正确运行。

通过以上的分析可以知道，要想成为一名优秀的程序员，必须具备多种不同的能力。