HZ BOOKS
华章教育

KNOWLEDGE · INNOVATION · CAPABILITY

高等院校计算机专业人才能力培养规划教材

C语言程序设计

王立柱 编著

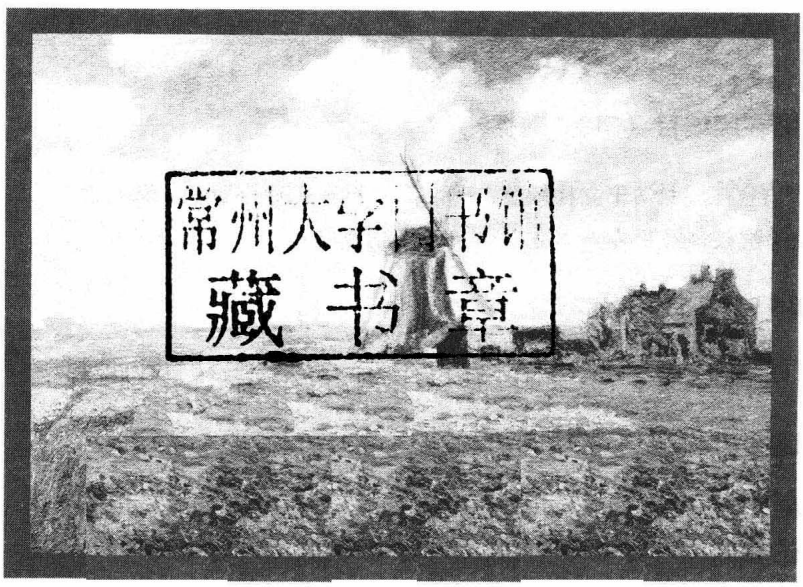
The C Programming Language

机械工业出版社
China Machine Press

高等院校计算机专业人才能力培养规划教材

C语言程序设计

王立柱 编著



The C Programming Language

机械工业出版社
China Machine Press

本书介绍利用C语言进行计算机程序设计的基本知识。全书共分12章，内容包括：机器语言简介，基本数据类型，操作符和表达式，程序流程控制，指针和数组，函数，模块化程序设计，字符串，结构、联合、枚举，流与文件，二维数组和指针以及C语言高级综合程序设计。

本书可作为大学本科C语言程序设计课程的教材，也可作为计算机编程爱好者的自学教材和参考书。

图书在版编目（CIP）数据

C语言程序设计 / 王立柱编著. —北京：机械工业出版社，2011.6
（高等院校计算机专业人才能力培养规划教材）

ISBN 978-7-111-34972-3

Ⅰ.C…　Ⅱ.王…　Ⅲ.C语言-程序设计-高等学校-教材　Ⅳ.TP312

中国版本图书馆CIP数据核字（2011）第105611号

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：李　荣
北京市荣盛彩色印刷有限公司印刷
2011年6月第1版第1次印刷
185mm × 260mm · 16.25印张
标准书号：ISBN 978-7-111-34972-3
定价：29.00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

出版者的话

机械工业出版社华章公司多年来以“全球采集内容，服务中国教育”为己任，致力于引进国际知名大学广泛采用的计算机、电子工程和数学方面的经典教材，出版了一大批在计算机科学界享誉盛名的专家名著与名校教材，其中包括Donald E.Knuth、Alfred V. Aho、Jim Gray、Jeffery D. Ullman等名家的一批经典作品。这些作品为我国计算机教育及科研事业的发展起到了积极的推动作用。

近年来，我们一直关注国内计算机专业教育的发展和改革并大力支持、参与相关的教学研究活动。2006年，教育部高等学校计算机科学与技术专业教学指导分委员会在对我国计算机专业教育现状和社会对人才的需求进行研究的基础上，发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》）。为配合《规范》的实施和推广，我们出版了“面向计算机科学与技术专业规范系列教材”。这套教材的推出，对宣传《规范》提出的“按培养规格分类”的理念、推进高校学科建设起到了一定的促进作用。

2007年，教育部下发了《关于进一步深化本科教学改革全面提高教学质量的若干意见》，强调高等教育以育人为本，以学生为主体，坚持以培养创新人才为重点，下大力气深化教育教学改革。在“质量工程”的思想指导下，各高校纷纷开展了相关的学科改革和教学研究活动。高等学校计算机科学与技术专业的教育开始从过去单纯注重知识的传授向注重学科能力的培养转型。2008年年底，教育部高等学校计算机科学与技术专业教学指导分委员会成立了“高等学校计算机科学与技术专业人才专业能力构成与培养”项目研究小组，研究小组由蒋宗礼教授（组长）、王志英教授、岳丽华教授、陈明教授和张钢教授组成，研究计算机专业人才基本能力的构成和在计算机专业的主干课程中如何培养这些专业能力。

为配合“高等学校计算机科学与技术专业人才专业能力构成与培养”专项研究成果的推广，满足高校从知识传授向能力培养转型的需求，在教育部高等学校计算机科学与技术专业教学指导分委员会专家及国内众多知名高校专家的指导下，我们策划了这套“高等院校计算机专业人才能力培养规划教材”。这套教材以专项研究的成果为核心，围绕计算机专业本科生应具有的能力组织教材体系。本套教材的作者长期从事教学和科研工作，他们将自己在本科生能力培养方面的经验和心得融入教材的编写中，力图通过理论教学及实践训练，达到提升本科生专业能力的目标。希望这些有益的尝试能对推动国内计算机专业学生的能力培养起到

积极的促进作用。

华章作为专业的出版团队，长久以来遵循着“分享、专业、创新”的价值观，实践着“国际视野、专业出版、教育为本、科学管理”的出版方针。这套教材的出版，是我们以教学研究指导出版的成功范例，我们将以严谨的治学态度以及全面服务的专业出版精神，与高等院校的老师们携手，为中国的高等教育事业走向国际化而努力。

华章教育

高等院校计算机专业人才能力培养规划教材

编 委 会

丛书序言

作为我国规模最大的理工科专业，计算机本科专业为国家的建设培养了大批人才。2006年，教育部计算机科学与技术专业教学指导委员会发布了《高等学校计算机科学与技术专业发展战略研究报告暨专业规范（试行）》（以下简称《规范》），提出了以“按培养规格分类”为核心思想的专业发展建议，把计算机专业人才划分为研究型、工程型、应用型3个类型。在《规范》的方针指导下，培养合格的计算机本科人才。

教育包括知识、能力、素质三个方面，专业教育不仅要重视知识的传授，更应突出专业能力的培养，实施能力导向的教育。如何以知识为载体实现能力的培养和素质的提高，特别是实现专业能力和素质的提高是非常重要的。对计算机专业本科教育而言，要想实现能力导向的教育，首先要分析专业能力的构成并考虑如何将其培养落实到教学实践中。为此，教育部高等学校计算机科学与技术专业教学指导分委员会开展了计算机科学与技术专业人才专业能力（简称计算机专业能力）的培养研究。该项研究明确了计算机专业本科人才应具有的4大基本能力——计算思维能力、算法设计与分析能力、程序设计与实现能力、系统能力，并将这四大基本能力分解为82个能力点，探讨如何面对不同类型学生的教育需求，在教学活动中进行落实。

为体现研究成果在教学活动中的实现，我们根据《高等学校计算机科学与技术专业人才专业能力构成与培养》，出版了这套教材。本套教材面向高等院校从知识传授向能力培养转型的需求，在内容的选择、体系安排和教学方法上按照专业能力培养的需要进行了探索。其主要特点有：

（1）以教学研究为先导。本套教材以计算机专业能力专项研究成果为基础，体现了先进的教育理念和教学方法，内容选择、知识深度、结构安排更加符合计算机专业教育的需求。

（2）落实能力培养的思想，同时满足课程的要求。本套教材不仅关注知识点的讲授，还凸显能力培养的要求，将能力的培养分解到各门课程的各个知识点的讲授中。

（3）力求贴近教学实际。作者均长期从事实际教学工作且对专业能力培养具有一定研究，教材编写注重科学组织内容、合理安排体系、便于教学实施，更具操作性。

（4）构建立体化教材。为了方便教师的教学活动，配合主教材开发配套的实验教材、教师参考书、学生辅导书、电子课件等教辅资源。

本套丛书的出版是在配合计算机专业能力的培养和落实方面的初步尝试，我们衷心希望本套教材的出版能起到抛砖引玉的作用，也希望广大教育工作者加入到能力培养的研究和实践中来，并对相关的教材建设提出自己的宝贵意见。

丛书编委会

前　言

畅行在程序和哲学之间

一、学习程序设计的方法

学习程序设计有两种方法：历史方法和逻辑方法。

历史方法是人们在不自觉中使用的方法。这种方法就是跟随事物的进程来分析和描述事物的方法。但是历史的进程常常是跳跃式的、曲折的，如果处处跟随着它，那就势必要关注许多无关紧要的、偶然性的东西，这些东西反映在思想意识上常常是相互冲突的，它们会不时地打断我们思想的进程，使我们很难形成前后连贯的、可以继承而深入的概念。而人的认知能力是有限的，不借助于前后连贯的概念，人们就很容易孤立、静止地看待问题，停留在通俗、肤浅的认识上。例如对不断出现的程序语言或程序设计的概念，有的人认为越新的越有价值，有的人认为越容易的越好，好像他们面对的是一堆彼此毫无关联的马铃薯，只有“新鲜的”或“好看的”才值得挑选。这种认识有多种表现形式，例如：有了C++就不必学习C了；有了Java就可以不学习C++了；“引用”是专门代替令人头疼的“指针”的；类概念的产生是因为我们眼前的事物都具有类的特征。他们喜欢从概念出发，忽视了用恰当的事实来说明概念形成的过程，不过这种认识是由历史方法决定的，因为事物的进程是跳跃式的、曲折的，不深入到事物的内部，就看不清形成这个概念的具体过程。

逻辑方法是本教材中采用的方法。这种方法要求把程序设计当作过程去研究，而且去除表面的、干扰性的、偶然性的因素，深入到程序设计的内在结构中去，研究它的发展规律。要知道“事物的表现形式和事物的本质不是直接合二为一的，否则一切科学都是多余的了”。要相信“精神本性也和肉体本性一样是必要的、具体的，并且同样具有严格的形式”，这种形式就是一系列不同的发展阶段，它们以一个否定另一个的方式彼此联系着。这里的否定不是消灭，而是扬弃，是后者克服前者中消极的东西，保留和继承前者中积极的东西，并把它发展到新的阶段，而连接各个阶段的是概念，概念既是对前一个阶段的总结，也是后一个阶段的起点。我们要做的就是由此及彼地推广这些概念，使它们互相隶属，从低级形式发展到高级形式。

逻辑方法要求我们至少做到五个方面：

1) 具体。要研究事实，对比事实，积累事实，用事实说话。要从程序设计的内部，即用程序设计的事实，来阐述程序设计的思想，不要从外部注入思想。例如，“类概念的产生是因为我们眼前的事物都具有类的特征”就不是用程序设计的事实阐述类的思想，而是在灌输类的思想，用这种思想不能说清一个简单的问题：我们眼前的事物早就具有类的特征，为什么

类的概念没有出现得更早一些?

2) 全面。要全面地看待事物，尽可能把握、研究它的一切方面、一切联系和中介。“具体之所以具体，是因为它是许多规定的综合，是多样性的统一。它在思维中表现为综合的过程，表现为结果，而不是表现为起点”。例如，引用的内部实质是指针，外部是“去引用”，形式等价于一个对象的别名。“‘引用’是专门代替令人头疼的‘指针’的”这种说法，仅仅是把引用看成了别名，而忽略了其实质。又例如，有人抱怨C++编译器暗地里对程序做了很多“手脚”，让程序员无所适从，这主要是因为他们还没有从程序设计的角度主动对待编译器这个连接高级语言和机器语言的中介。

3) 过程。“要从事物的发展和变化中来观察事物”。“结论要是没有使它得以成为结论的过程，就毫无价值；结论本身若是固定不变，若是不再成为继续发展的前提，就比无用还糟糕”。用一句话来表达，就是过程比结果更重要。这是新的唯物主义观点的直接的理论前提，也是逻辑方法的出发点。例如，有人认为可以直接学习C++，而不用先学习C。实际上，学习C++也要先学习基本类型、操作符和表达式、程序流程控制、指针和数组、函数、C格式字符串等，而这些绝大部分都是C的内容，即使是那些与C不同的部分，也需要“从C基本类型到C++用户类型再到C基本类型”这个否定之否定的过程来给予说明。例如，C++基本类型变量的初始化形式如下：

```
int n(5);
```

C++从左至右的联合赋值表达式如下：

```
(m=n)=5;
```

C++的函数重载、内联函数和默认参数，都是C中没有的内容，但是如果我们尊重它们产生的过程，就应该在C++用户类型和C基本类型如何统一的阶段中来学习，这样一来，直接学习C++和先学习C再学习C++就是一回事了。

4) 相互作用。事物是相互作用的，在大多数情形下，正是忘记了相互作用，所以阻碍了我们看清最简单的事物。例如，很多人都不是把存储和算法的相互作用和不断发展的关系当作程序设计方法，而是把诸如顺序、循环和选择等程序流程控制结构当作程序设计方法，认为任何程序都离不开这些结构，因此它们是方法。可是这种适用于任何情形的方法在像字符串匹配这样的算法设计中，其作用却微乎其微，甚至无用武之地，因为用C++实现仅是一条函数调用语句，用C实现需要模块化设计，用机器语言实现就要熟悉计算机组成和工作过程了。

5) 实践。实践是不断遇到问题和解决问题的过程，是认识的源泉，或者说不断遇到问题和解决问题的过程是改造世界的过程，人首先是改造世界的主体，然后才是认识世界的主体，这是实践主体性。因此，“必须把人的全部实践、人的需要，包括到对事物的完美的定义中去”。“凡是把理论导致神秘主义方面去的神秘东西，都能在人的实践中以及对这个实践的理解中得到合理的解决”。用Stroustrup的话讲：C++的每一步演化和发展都是由于实际问题所引起的。它的任何概念都不是卓越的个人苦思冥想的结果。虽然有不少语言是为了证明一个观点而设计的，但却是C++成为系统设计的主流语言。

具体、全面、过程、相互作用和实践，这些虽然不是逻辑方法的全部，但就我们目前的

学习来说，已经足够了。我们知道，C++标准模板库的出现，特别是常用的string、vector、list和适配器这些常用的部分，使学习C++和学习数据结构的顺序发生冲突：学习数据结构需要C++，而C++中又包含着数据结构，它们互为前提。合理地解决这个问题，需要综合运用逻辑方法。先考虑C++编译器这个中介，它更多地是把C++代码转换为C代码，然后利用C编译器把C代码转换为机器代码。如果了解从C++到C的转换过程，就不会抱怨C++编译器暗中“做手脚”。再考虑C++标准模板库，string、vector、list和适配器在很长时间里一直属于C描述的数据结构，这不是一个可以转过头说个“不”就可丢弃的程序设计实践。C++标准模板库告诉我们，这里包含着很多对C++的必然性的思考，我们的任务就是把它们精炼出来，和C++的描述对比，用前者到后者的转换过程来表达这个思考，这不是用三两句话就可以表达出来的。科学成就在本质上是积累的结果，科学是继承性最强的文化形态之一，而不用逻辑方法，积累是不可能的。

二、丛书的内容

运用上述逻辑方法，我们进行了长期的包括C、C++和数据结构在内的教学实践，收效显著。现在我们把积累的经验和对逻辑方法更系统、深入的理解及运用都融入到一套程序设计系列教材中。这套教材包括：《C语言程序设计》、《C++语言程序设计》（基于C，作为第2门程序设计课教材）、《数据结构：C++语言描述）》、《C++语言程序设计》（从过程化到面向对象，作为第1门程序设计课教材）、《Java语言程序设计》和《C和C++程序设计中的100个难题》。高校教师可根据自己学校的情况选用。

我们选择C、C++和Java，是因为它们不仅是系统设计的主流语言，而且也代表着程序设计的发展；我们选择数据结构，是因为它不仅代表着程序设计的主要方法，而且STL的出现，使数据结构和程序语言达到了几近完美的结合。它们结合在一起，为我们认识、学习、运用和丰富逻辑方法提供了广阔的空间。

上述逻辑方法的思想，大多引自我们编写的《马克思恩格斯哲学原著英汉对照选读》一书。编写这本书的目的是为了更系统地掌握辩证唯物主义思想，加强我们教学的原则性和系统性。这本书的第2版正在准备中，新版书可以作为这套程序设计课程群教材的教师参考书。

三、本书的脉络

本书以程序设计的发展过程为对象，以程序设计的基本矛盾的运动为主线，把程序设计的需要作为原则，用程序设计的实例来贯穿。

什么是程序？程序是在某种存储模式上实现的算法，即程序=存储模式+算法。所谓存储模式是指数据和对数据的基本操作的存在形式。所谓算法是指可以用基本操作来表示的求解问题的有穷序列。算法好比一台机器，构造再复杂，从力学上看，也不过是简单机械力的不断重复。最先提出这个思想的人是图灵，他提出，应该用机器保留一些最简单的操作，然后将一个复杂的计算分解为这些操作。实现这个思想的代表人物是冯·诺依曼，他用固化在硬件中的机器指令表示简单操作。这些机器指令的集合构成的系统成为第一个程序语言——机

器语言。从此，存储模式都用程序语言来表示，程序也就成为用某种程序语言实现的算法，例如机器语言程序、C语言程序、C++语言程序、Java语言程序等。

什么是程序设计？程序设计就是程序设计方法。“纵观短暂的计算机发展史，存储和算法这两个方面一直存在，发展演化的只是它们之间的关系，这个关系就是程序设计方法”。这个关系的发展经历了这样几个阶段：基于机器指令系统存储模式的过程化程序设计、基于基本类型存储模式的类型化程序设计、基于复合类型存储模式的过程化程序设计、基于抽象类型存储模式的程序设计、基于类存储模式的对象化程序设计、基于继承和动态绑定的面向对象的程序设计、基于标准模板库的泛型程序设计。其中，基于抽象类型的高级程序设计就是数据结构。机器语言（或者汇编语言）、C语言和C++语言与这个过程是对应的。

什么是程序设计的基本矛盾？程序设计的基本矛盾是存储和算法。从基于机器指令系统的过程化程序设计到基于标准模板库的泛型程序设计的一系列相互连接的发展阶段，都是由这个矛盾推动的。

基于上述基本思想，本书的内容安排如下：

第1章简述基于机器指令系统存储模式的过程化程序设计，主要内容有：

1) 一个求和程序，说明存储模式与计算机组成及工作过程的联系。

2) 一个方程求解的两种程序设计方法，说明存储模式和算法的相互依赖关系。

3) 综合分析存储模式和算法的矛盾，并给出通向下一阶段的途径。

第2章～第4章讲授基于基本类型存储模式的类型化程序设计，主要内容有：

1) 从数值算法的基本需要出发，介绍各种固有基本类型存储模式的特点。

2) 操作符表达式与基本类型的统一。

第5章～第7章讲授基于指针与数组复合类型的过程化程序设计，主要内容有：

1) 从非数值算法的基本需要出发，描述数组和指针应有的特点。

2) 从地址到指针的类型化过程。

3) 数组的两种定义格式和对应的两个方面的特征——数组变量和数组指针。

4) 数组和指针的相互依存关系。

5) 函数与操作符表达式的统一：函数是操作符表达式的逻辑延伸。

6) 函数调用和赋值操作的统一：函数调用过程可以概括为三步，每一步都是赋值操作。

7) 递归和非递归函数的统一：递归和嵌套调用的统一；递归和递推的统一；不同层次递归和函数的统一。

8) 过程化程序设计的规模化——模块化程序设计。

第8章～第10章讲授基于抽象存储模式的程序设计，主要内容有：

1) 基于系统支持的抽象存储模式——字符串。

2) 用自定义函数代替系统提供的字符串基本操作函数。

3) 用自定义函数补充字符串的不足。

4) 基于自设计抽象存储模式的程序设计——Date结构。

5) 基于系统支持的抽象类型存储模式的程序设计——文件。

第11章从二维数组和指针出发，对各种数组和指针的概念做了综合性分析和概括。

第12章讲授基于系统支持的抽象类型存储模式的高级程序设计和基于类存储模式的对象化程序设计，主要内容有：

1) 顺序表。

2) 链表。

3) 结构串（简介）。

4) 综合设计——文本统计。

5) 分析C语言的局限性。

6) 基本内容从C到C++。

7) 顺序表从C到C++。

四、教学经验和体会

1. 为什么学习程序设计

为什么要学习程序设计？这是首先要向学生说明的事情。我们一般都会告诉他们：计算机已经深入到我们的生活，并不断影响和改变着我们的生活，学习程序设计可以更深入地理解和更有效地应用计算机。这样回答是没有什么问题的，但是还远远不够，学习程序设计更是为了学习逻辑方法，提高科学素质，尽快完成从中学到大学的角色转变。

逻辑方法要求把事物的发展过程当作对象，深入研究它的发展规律，这种方法在中学是不学的，而在大学是重点学习的方法。要学好逻辑方法必须将实验和专业课程结合起来学习，计算机的广泛应用，使程序设计特别适合于这个专业课程角色。当然，不能说只要学习程序设计就能学会逻辑方法，只有深入到程序设计的内在结构中去研究它的发展规律，才能达到我们的学习目的。

科学素质是指分析和处理复杂问题的能力。所谓复杂，是指包含着很多必然联系的因素。所谓能力，是指能够认识到并把握住这些因素之间必然的联系。要提高这种能力，就要从解决实际的复杂问题的过程中学习。“C++的设计就是为了使较大的程序能够以一种合理的方式构造出来”，本教材实现了从C到C++的无缝连接。

作为老师，不要仅仅从实用的角度或仅仅为了应试来讲授程序设计，这会使课程支离破碎，只剩雕虫小技。要讲逻辑，注重整体，前后连贯地讲授知识，使学生具备全局观、系统观，这是大学老师的义务、责任和价值。

2. 培养学习兴趣，树立学习信心

比教学生编程技巧更重要的是培养他们学习程序设计的兴趣，树立他们的自信心。编程技巧是从模仿开始的，模仿什么和怎样模仿取决于学习什么和怎样学习，而后者就是学习方法。学习兴趣是由学习什么和怎么学习来决定的、能够提高学习者认知能力和自身价值的、可持续的知识积累过程。自信心是对这个过程的认同。而这个过程与程序设计的发展过程是统一的。

我们之所以欣赏计算机的美，不只是因为它具有令人感叹的人类智慧的美，更是因为我们通过比较全面而深刻地认识它内在的变化规律，能够比较全面而深刻地挖掘出自己的潜能，

最终欣赏到自身的美。

3. 因材施教

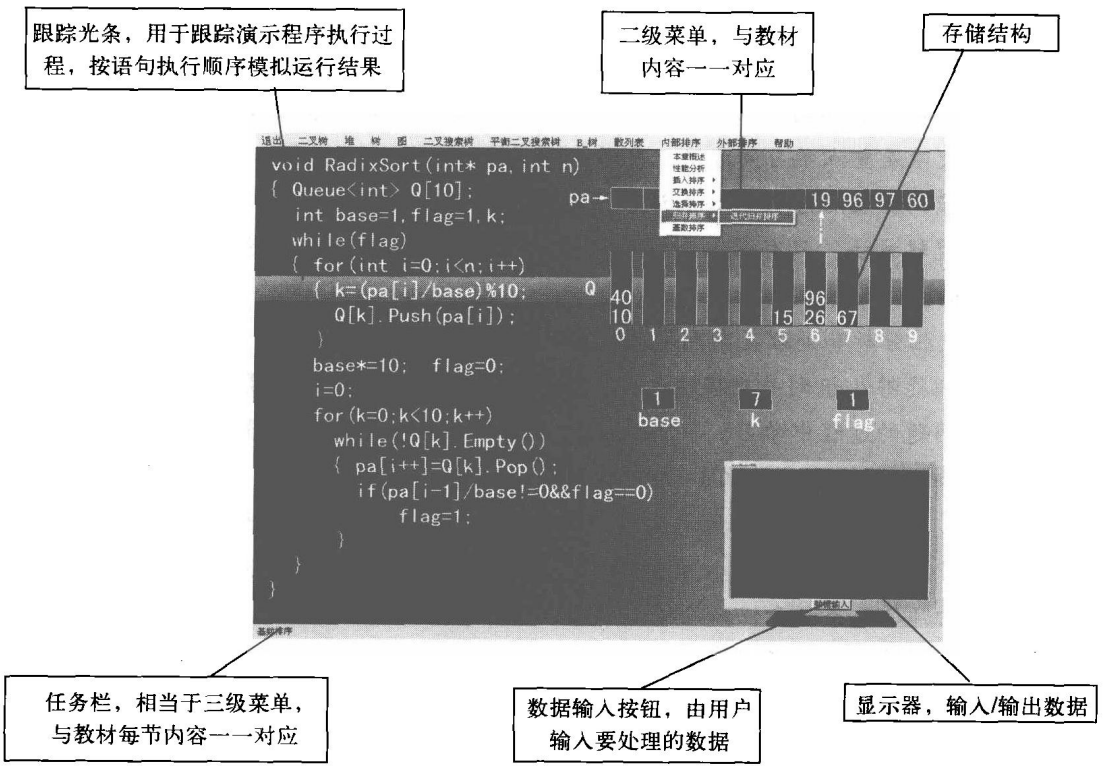
本书如果作为计算机专业的教材，总学时数不少于80，那么每一章中的深入讨论部分和第12章都应该讲。

本书如果作为非计算机专业的必修课教材，总学时数不少于60，深入讨论部分、8.5节和第11章可以省略或要求自学，第12章可以作为综合设计的实训内容。

如果总学时数少于60，就把重点放在第5章、第6章和第8章的前四节。

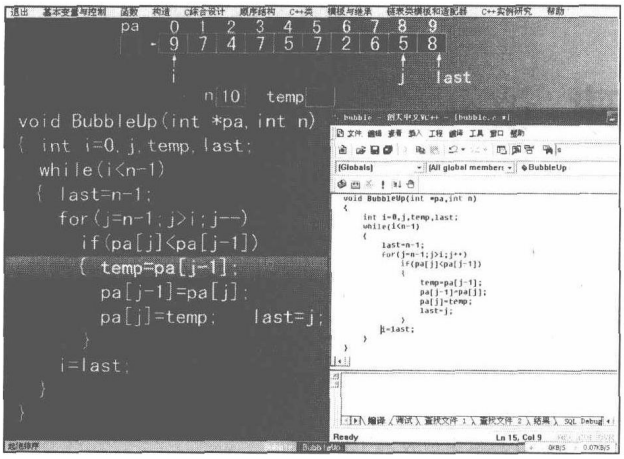
4. 多媒体教学

本书有配套的多媒体软件，既助学又助教，使结构、算法、代码、运行过程和结果同时展现、数据自由输入、运行过程步步跟踪。它适应了传媒时代的要求，扩展了学生自主学习的空间，解决了算法复杂抽象、学生难以理解、教师难以讲解的难题，加强了师生的切磋交流，为教师前后横向对比、全面具体、综合系统、深刻展示思想方法地授课搭建了平台。本书配套的学生用助学软件、教师用教学软件以及源代码均可从华章网站（www.hzbook.com）下载。另外，该多媒体软件是基于Authorware应用软件的编程特性开发的，需要有很好的程序设计基础和综合能力，它从一个方面时时彰显着学习程序设计的意义（如下图所示）。



5. 实验

程序设计类课程注重实验，可是实验课时普遍不足，甚至可能因为缺少教师而使教师难以关照每一个学生。多媒体课件可以在一定程度上代替教师讲解很多实验中遇到的问题，而且可以把学生分组，每组由优秀的学生担任组长，遇到问题时，先对比课件，集体讨论（如下图所示）。



6. 考试

考试要突出系统性和逻辑性知识，树立学生的全局观和发展观，切忌偏题、怪题、随意编造题。指针和函数是C语言的核心内容，也是在程序语言中不断发展的核心概念，因此考试要围绕这两点系统地展开。题目可以公开，在几套题中随机选取，概念题笔试，程序题上机编写和调试，其中有常规题和自选题。就像体操和跳水比赛一样，动作和难度系数公开，有规定动作和自选动作。下面是一套针对计算机专业的期末考题：

1) 笔试部分——举例说明或论述下面的概念或题目：

- 地址和指针的关系。
- 数组变量和数组指针。
- 数组和指针相互依存。
- 数组变量指针和数组指针。
- 二维数组变量、二维数组指针、行数组变量和行数组指针的区别和联系。
- 二维数组、一维数组、二维指针变量、一维指针、指针的指针、一级和二级指针的区别和联系。

2) 编程部分——上机编写和测试下列算法：

- 用自定义函数代替系统提供的字符串基本操作函数，并编写应用程序检验。
- 用自定义函数补充字符串的基本操作函数，并编写应用程序检验。

五、致谢

感谢武警医学院贾冬梅老师，她对这本教材仔细校对了两次，不仅指出很多文字上的错误，而且提出不少建设性的意见，使我几次在内容和布局上做了调整。

感谢天津师范大学的刘志红老师，她不仅是课程改革团队的主要成员、教学课件的制作者、课程的主讲教师，而且十年来，她的学生评教成绩一直很优秀，名列前茅。要知道，我们的课程包括C、C++和数据结构，讲授两个学期，每届学生有几百人，在这样的教学规模下，能够有如此的成绩，不仅对我，而且对我们周围很多从事计算机教育的老师，都是一种鼓舞。我们所遵循的逻辑方法不仅经过她的教学实践证明是普遍受学生欢迎的，而且也通过她的实践得到不断的完善。

限于作者水平，书中难免有不足或不当之处，恳请读者、同行批评指正。

作 者

教学建议

教学章节	教学要求	课时 （授课课时+上机课时）
第1章 机器语言简介	了解计算机组成及其工作过程 了解机器语言程序——例1.1和例1.2 了解存储和算法是一对矛盾体	1+1
第2章 基本数据类型	掌握C程序的基本构成 掌握整型、字符型和实型常量的基本操作和输出格式 了解字面值常量的意义和左右值概念	3+3
第3章 操作符和表达式	掌握表达式的构造 掌握各种表达式的基本内容 掌握内部类型转换	2+2
第4章 程序流程控制	掌握选择结构、for语句，讲解程序4.1~4.9	2+2
	掌握while、do-while语句，讲解程序4.10~4.15 掌握循环嵌套，讲解程序4.16和4.17 掌握break和continue语句，讲解程序4.18和4.19	2+2
第5章 指针和数组	掌握指针和数组的概念 掌握指针和数组的相互依赖关系，讲解程序5.1~5.6	2+2
	掌握指针和数组的应用设计举例 了解指针、数组、表达式和左值的关系 了解数组变量指针和数组指针的关系	2+2
第6章 函数	掌握函数定义与调用、值传递和指针传递以及函数声明（原型），选讲程序6.3~6.8	2+2
	掌握变量作用域和变量生命周期的概念 掌握自动局部变量和静态局部变量 了解外部变量和寄存器变量 掌握动态空间管理	2+2
	掌握函数指针、递归 掌握输入/输出函数 了解函数的深入探讨中的内容并自学掌握	2+2
第7章 模块化程序设计	了解全局外部函数、静态外部函数、全局外部变量、静态外部变量 掌握const型变量、间接const型指针、const型指针、全const型指针的概念和用法 掌握编译预处理的各种指令 自学掌握模块化应用设计举例	2+2
第8章 字符串	掌握字符串常量和赋值、字符串基本操作函数 选讲和掌握字符串基本操作函数的补充内容，自学完成剩余部分 了解字符串的深入探讨中的内容	2+2

（续）

教学章节	教学要求	课时 （授课课时+上机课时）
第9章 结构、联合、枚举	掌握结构的定义和赋值、结构数组、结构嵌套 掌握结构返回值和指针传递的关系 了解联合和枚举概念 掌握结构应用设计举例中的Date结构，其他应用设计自学	2+2
第10章 流与文件	掌握文件的各种读写操作，选讲程序10.1~10.6，其余自学	2+2
第11章 二维数组和指针	掌握二维数组与二维数组指针、二维数组与一维数组的关系、指针数组与二级指针变量的关系、指针数组与二维数组的关系 了解二级指针指向的动态数组空间、以二级指针为参量的main函数 了解马鞍点程序设计 了解一级和二级指针与一维和二维指针的关系	2+2
第12章 高级综合程序设计	掌握顺序表和单向链表，其他内容辅助自学	6+6
总课时	第1~12章建议课时	36+36

目　录