21世纪

高等学校精品规划教材

案例式教学

C/C++程序设计

李云峰　李　婷　编著

中国水利水电出版社
www.waterpub.com.cn

21世纪高等学校精品规划教材

C/C++程序设计

李云峰　李　婷　编著

中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书针对初学者的特点，采取“提出问题→分析问题→解决问题→归纳提高”的四部曲教学模式，分为10章。主要内容包括C语言程序设计概述、数据类型与运算、结构化程序设计、利用函数编程、利用数组编程、利用指针编程、利用构造类型编程、文件操作、C++程序设计、综合应用程序设计。

本书分为3个层次，基础层介绍程序设计的基本概念、C基础与程序结构，给学习者建立一个全面的程序概念；提高层介绍C语言程序的基本设计方法、算法案例程序设计；引申层介绍C++程序设计和综合应用程序设计。

本书定位准确、结构合理、内容翔实、概念清晰、逻辑性强、例题丰富、深入浅出、循序渐进、通俗易懂，符合学习者的认知规律，并有配套的《C/C++程序设计学习辅导》（其内容包括编程指导、习题解析、实训指导、知识拓展），是程序设计学习者的理想用书，还可作为大学理工科类C语言设计或程序设计基础课程教材，也可作为大学生程序设计竞赛的基础训练教材。

本书配有免费电子教案，读者可以从中国水利水电出版社网站以及万水书苑下载，网址为：http://www.waterpub.com.cn/softdown/或 http://www.wsbookshow.com。

图书在版编目（CIP）数据

C/C++程序设计 / 李云峰，李婷编著. -- 北京 : 中国水利水电出版社，2012.8
21世纪高等学校精品规划教材
ISBN 978-7-5170-0002-0

Ⅰ. ①C… Ⅱ. ①李… ②李… Ⅲ. ①C语言－程序设计－高等学校－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第173338号

策划编辑：雷顺加　　责任编辑：李　炎　　封面设计：李　佳

书　名	21世纪高等学校精品规划教材 **C/C++程序设计**
作　者	李云峰　李　婷　编著
出版发行	中国水利水电出版社 （北京市海淀区玉渊潭南路1号D座　100038） 网址：www.waterpub.com.cn E-mail：mchannel@263.net（万水） sales@waterpub.com.cn 电话：（010）68367658（发行部）、82562819（万水）
经　售	北京科水图书销售中心（零售） 电话：（010）88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排　版	北京万水电子信息有限公司
印　刷	三河市铭浩彩色印装有限公司
规　格	184mm×260mm　16开本　23.25印张　586千字
版　次	2012年8月第1版　2012年8月第1次印刷
印　数	0001—3000册
定　价	38.00元

前　言

随着信息技术的迅速发展和广泛应用，程序设计已成为高校理工学科各专业普遍开设的一门公共基础课。而C语言作为程序设计的主流语种，经历了30多年的发展和不断完善，已成为国内外公认的一种优秀程序设计语言，日益显示出其它语言不可比拟的优点，因而也成为理工学科普遍开设的一门程序语言课程。那么，如何提高该课程的品质以满足学习者的需求，是需要我们不断探索的课题。

目前，C/C++语言课程教材可分为两类：一类是以语法学习为中心，在介绍语法的基础上，结合程序设计巩固语法知识，强调的是语言知识的掌握而不是程序设计能力训练。另一类是案例式教材，以案例分析为主兼顾语法教学，引导学生通过模仿学习程序设计，强调的是应用程序设计方法的掌握。相对前一类，后一类教材语法知识的完整性和程序设计知识的系统性(算法、数据结构等)相对欠缺。

本教材便是基于这一理念并总结“C/C++语言程序设计”课程教学改革实践和精品课程建设实践的基础上编写的。我们认为，程序设计不仅要让学生掌握扎实的语法知识，而且应当在这个基础上重点培养学生的编程能力和创新思维能力。创新思维的培养是潜移默化的，教材应当在创新思维方面加以引导，培养学生发现问题、分析问题、解决问题的能力，构建合理的知识与能力体系。本教材力求体现以下特点：

1. 以问题需求为引导，激发学习的主动性。从实际应用需求（该章的教学目标）出发，通过“问题原由”，引出该章所要讨论的问题和所要掌握的知识内容；通过“问题描述”引出该节的知识要点和语法结构。以此激发学生学习的主动性和求知欲，避免学习的盲从性，消除语法规则学习的枯燥感，克服程序设计难学的心里障碍，达到学用结合的目的。

2. 以程序实例为主线，注重创新思维培养。教材贯穿着提出问题、分析问题、解决问题的思路，对每一个知识点都给出了实例程序，通过“解题分析”，引导学生思考解决问题的方法与路径；通过“程序实现”，归纳出解决问题的方法和步骤；通过“问题点拨”，引导学生触类旁通。从而，培养学生的创新思维与分析问题和解决问题的能力。

3. 以算法案例为核心，突出实用性和趣味性。本教材每章均以“算法案例程序设计”概括本章的教学内容，将程序规则与算法分析相结合。教材中的算法案例大都选用具有理论研究价值、实际应用价值和颇有趣味性的经典算法，以此培养学生对程序设计的学习兴趣，提高实际编程能力，从而体现了即学即用，学用结合的原则，避免了学而不用，或会学不会用的问题。

事实上，作为程序设计语言教材，如何处理程序设计语言与算法的关系是极为重要的。Pascal语言设计者、“图灵”奖得主Niklaus.Wirth教授曾提出了一个著名的论断：程序＝算法＋数据结构。这个论断的要旨是：程序的核心是算法，算法的本质是处理数据，算法与数据不可分离。本教材充分体现了这一论点，精心策划算法与数据结构的知识布局，将穷举算法、迭代算法、递推算法、递归算法、回溯算法、贪心算法、动态规划、运算模拟、排序算法、查找算法、线性表、队列、栈、树、图等知识，以知识拓展的形式合理安排在辅助教材各章中，与主教材中的“算法案例程序设计”相对应。这些算法知识为设计具有专业水准的应用程序奠定

了良好的理论基础。

4. 以工程应用为目标，强化综合应用能力培养。教材第 10 章“综合应用程序设计”以工程项目的形式，将每一个项目设计成：系统设计目标、系统需求分析、系统总体设计、系统功能实现等四个环节，以此提高学生运用所学知识开发应用软件的综合设计能力，引导学生养成良好的设计思路和编程习惯，积累编写和调试大型程序的经验，编写出风格优美、可读性强、易于维护的程序代码。

为了强化综合应用能力培养，充分体现以教师为主导，以学生为主体的教学思想，本课程的教学设计（教学手段、教学内容、教学方式和教学资源等方面）力求做到 3 个结合：

（1）课堂教学与自主学习相结合：我们编写了与主教材配套的《C/C++程序设计学习辅导》，其内容包括编程指导、习题解析、实训指导和知识拓展（常用算法、数据结构和相关问题概念）。这样，既节省课堂教学时数，又最大限度地为学生自主学习提供方便；既为学好程序设计提供强有力的支持，又为学习者提供上升与拓展的空间。

（2）纸质教材与电子资源相结合：配合纸质教材，我们建有课程教学网站，内容包括授课文本教案、电子教案（PPT）、在线测试、答疑解惑、视频教学等，使本课程融教、学、做为一体，集多种媒体于一身，为学习者提供可选择的学习资源与全方位的学习支持服务。

（3）教学内容与等级考试相结合：教材内容涵盖了全国计算机等级考试大纲二级 C 语言程序设计考核的全部内容，并将近年来的考试试题编入到习题和实训内容之中，以便学生参加国家 C 语言程序设计等级考试，增强水平等级考试能力，激发和提升学习动力。

教材编为 10 章，分为 3 个层次：第 1~3 章为基础层（C/C++程序设计概述、数据类型与运算、结构化程序设计）；第 4~8 章为提高层（利用函数编程、利用数组编程、利用指针编程、利用构造类型编程、文件操作）；第 9~10 章为引申层（C++程序设计、综合应用程序设计）。

本教材采取进阶式结构、案例式引导、解析式分析，从而使本教材教学目标明确、结构清晰、循序渐进、逻辑性强、实用性好。通过潜移默化的引导，激发学生的学习兴趣，开发学生的创造潜能，培养学生自主学习的意识，营造以学生为主体的学习氛围，达到提高学生综合应用能力的目的。

本教材既适合作为高等院校程序设计课程的教材，也适用于全国计算机等级考试二级 C 语言程序设计培训教材或 C 语言程序设计竞赛基础训练教材。同时，也适用于 C 语言程序设计爱好者自学用书。

本教材由李云峰教授和李婷博士(副教授)编写。曹守富老师为本书程序调试、课程网站建设做了大量工作，丁红梅、周国栋、刘艳、刘冠群、谭阳、方颂、陆燕等老师参加了课程教学资源建设。在编写过程中，参阅了大量国内外同类优秀教材和专著，并从中吸取了许多有益的营养，特别是湖南理工学院杨克昌教授，为本书提供了极为珍贵的文献资料，在此，谨向这些著作者一并表示衷心感谢！

本书凝聚了作者多年教学、科研以及软件开发的经验和体会，尽管我们希望做到更好，但因作者水平所限，书中难免存在许多不足之处，敬请专家和读者批评指正。

作 者

2012 年 6 月

目　　录

导　　学

C/C++语言是出色的程序设计语言，它以语言精练、使用灵活、结构清晰、功能强大、可移植性好、应用领域广泛而著称。虽然程序设计语言种类繁多，而且各具特色，但C/C++语言一直作为软件编程者的首选和高等学校程序设计课程的教学语言。

那么，如何教好、学好C/C++课程呢？课程导学的意义形如一个人到了一个陌生的城市，如果先站在该城市的最高处俯视整个城市，就会对该城市有个大致了解。对一门课程的学习也是这样：

（1）了解该城市的基本布局（本课程的知识结构）

（2）了解该城市的交通线路（本课程的教学主线）

（3）了解该城市的主要建筑（本课程的主要内容）

（4）了解该城市的外围环境（与其他课程的关系）

我们希望通过“课程导学”，能为使用本教材的教师和学生提供有益的参考。

一、C语言程序设计

1. 学习C语言的意义

现在可供选择的程序设计语言很多，并各有其特点和适合应用领域，但在工程上和教学中使用最多的是C语言。C语言之所以能在众多的高级语言中立于不败之地并一直处于发展之中，是因为C语言功能丰富、表达能力强、使用灵活方便、应用面宽、目标程序效率高、可移植性好。它既具有高级语言的优点，又具有低级语言的许多特点；既适合编写应用程序，又能编写系统软件。因此，C语言在工程应用与教学方面都有其不可替代的优势。

在工程方面，C语言应用越来越广。例如在单片机系统中，C语言逐步取代原来的汇编语言，这里包括销量很大的Intel系列单片机、Motorola系列单片机和Philips系列单片机。

在教学方面，C语言已替代了过去的Pascal语言，成为高等院校计算机专业以及其他一些工科类专业学生必修的一门计算机语言课程。现在计算机专业学生必修的“数据结构”以及“操作系统”课程基本上都以C语言作为先修课程，因为这两门课程中的算法描述和程序设计大多使用C语言。因此，学好C语言是非常重要甚至是必需的。事实上，只要掌握了C语言，就可以毫无困难地学习并掌握其他程序设计语言。

或许有人认为C++语言的出现，意味着C语言已经过时或淘汰。其实，这是一种误解。C++是为处理较大规模的程序开发而研制的大型语言，它比C语言复杂得多，难学得多。特别是对于初学者而言，很多抽象概念，一时难以掌握。事实上，并不是每个人都需要用C++编制大型程序。相对而言，C语言更适合于解决某些小型程序的编程。C语言作为传统的面向过程的程序设计语言，在编写底层的设备驱动程序和内嵌应用程序时，往往是更好的选择。

由于C语言具有功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好等特点，不仅适用于开发应用软件，还适用于开发系统软件，因而深受程序设计者的青睐。一旦掌握了C语言，就可以较轻松地学习其他任何一种程序设计语言，并能为今后的专业学习或软件开发打下良好的基础。因此，现在大多数高校仍然把C语言作为第一门计

算机语言进行程序设计教学。事实上，学好 C 语言，能为学习 C++、Java、C＃等现代程序设计语言奠定良好的基础。

美国一位资深软件专家写了一篇文章，题目是“对计算机系学生的建议”。他说“大学生毕业前要学好 C 语言，C 语言是当前程序员共同的语言。它使程序员互相沟通，比你在大学学到的‘现代语言’（比如 ML 语言、Java 语言、Python 语言或者正在教授的流行语言）都更接近机器”。他指出：“不管你懂得多少延续、闭包、异常处理，只要你不能解释为什么 while（*s++=*t++）的作用是复制字符串，那你就是在盲目无知的情况下编程，就像一个医生不懂最基本的解剖学就在开处方”。

本教材在介绍 C 语言的基础上，在第 9 章以对比的方式介绍了 C++的基本概念及其简单应用，这样很有利于对 C++的理解，为深入学习 C++打下良好基础。

2. 程序设计的内涵

（1）程序设计是能力培养的起点：计算机的本质是“程序的机器”，程序和指令的思想是计算机系统中最基本的概念，是冯• 诺依曼计算机的核心。因此，只有懂得程序设计，才能进一步认识和应用计算机。通过学习程序设计，能够进一步了解计算机的工作原理，掌握用计算机处理问题的方法，培养分析问题和解决问题的能力，锻炼和提高逻辑思维能力。

（2）算法是程序设计的灵魂：计算机在解决各个特定任务中，通常涉及两个方面的内容——数据和操作。所谓“数据”，是指计算机所要处理的对象，包括数据类型、数据的组织形式和数据之间的相互关系（数据结构）；所谓操作，是指计算机处理数据的方法和步骤（算法）。它们是利用计算机解决实际问题时首先要掌握的思路和方法。1974 年图灵奖的获得者、著名计算机科学家、算法大师克努特（Donald E.Knuth）说：“计算机科学是算法的学习”。瑞士著名计算机科学家、Pascal 语言的发明者尼克莱斯 • 沃斯（Niklaus Wirth）教授早在 1976 年提出了这样一个公式：

算法+数据结构=程序

由此可见，算法、数据结构和程序，三者密不可分。程序的目的是加工数据，而如何加工，则是算法（对数据的操作）和数据结构（描述数据的类型和结构）的问题。在加工过程中，只有明确了问题的算法，才能更好地构造数据，但选择好的算法，又常常依赖于好的数据结构。事实上，程序就是在数据的某些特定的表示方式和结构的基础上对抽象算法的具体描述。因此，编写一个程序的关键是合理组织数据和设计好的算法。

（3）语言是程序设计的工具：程序设计的另一个问题是怎样实现算法，即用计算机语言编写程序，达到用计算机解题的目的。学习程序设计语言的目的有 2 个：

① 学习用计算机语言编写程序，即掌握程序设计语言中的语法规则。

② 学习程序设计方法及其算法，即掌握利用计算机解决实际问题的思路和方法。

在学习过程中，不仅要要正确处理好程序与算法的关系，还要处理好程序与语言的关系。在接受一个任务后，首先要考虑解题的思路，即设计算法，然后考虑怎样编写成程序。算法和语言两者都很重要，不掌握算法则不可能编出高质量的程序，不掌握程序设计语言则谈不上编写程序。

二、课程教学定位

1. 课程性质

“C/C++程序设计”带有基础学习性质，而且是一门操作性很强的课程。通过该课程的学

习，可以进一步加深对计算机基本概念的理解；训练逻辑思维能力；掌握正确和规范的编程方法；培养严谨的科学作风；获取解决问题的方法和思路。

本课程的先修课程有高等数学、计算机电路基础、计算机导论或大学计算机基础等。

2. 课程特点

C/C++语言是一门实践性很强的课程，必须通过大量的编程实践和上机调试，方能掌握。有人说，计算机语言课程是三分教、七分练。事实上，即使熟练的程序员设计的程序，同样需要上机调试通过。由此可见，计算机语言课程实践性很强的重要特点。

3. 课程目标

程序设计是高等学校计算机或相关专业一门重要的必修课程，通过该课程的学习，要求达到三个基本目标：一是使学生在掌握程序设计语言知识的同时，学会使用计算机处理问题的方法，培养学生思考问题、分析问题和解决问题的能力；二是在掌握程序设计方法的同时，进一步加深对计算机工作原理的认识，懂得怎样让计算机按照人们的意图进行工作；三是通过程序设计，培养学生逻辑思维、抽象思维的思想和方法，使学生具备程序设计的能力。

4. 课程任务

C/C++程序设计课程蕴含了丰富的程序设计理念，通过本课程的教学，使学生熟练掌握C/C++语言的基本语法规则，写出语法正确的程序，并养成良好的程序设计风格；掌握结构化程序设计方法，会使用算法的描述工具（如自然语言、程序设计语言、伪代码、流程图、N-S结构图、PAD结构图等）；掌握最基本的算法（如穷举、最大最小值、查找、插入、删除、排序、迭代、递推等）；培养学生调试程序的能力，即能在Turbo C++ 3.0或Visual C++ 6.0的集成环境下，对程序进行编辑、编译、调试和运行等。通过本课程学习，基本具备开发应用程序的能力；同时，为数据结构、软件工程等后续课程学习打下坚实基础。

三、C/C++程序设计流程

程序是对计算任务的处理对象和处理规则的描述，处理对象是数据或信息，处理规则一般指处理动作和步骤。程序设计是设计、编制和调试程序的方法与过程。为了说明什么是程序设计，下面通过两个实际例子，以此引入程序设计的基本概念。

【实例1】设有如图1所示的五边形，给出五边形的边及对角线的长度，如何求取该多边形的面积呢？

（1）分析问题：本例求五边形的面积，事实上，只要求出三角形S1、S2、S3的面积之和即可。

由于要三次计算三角形的面积，为了程序的简单起见，可将计算三角形面积定义成函数，然后在主程序中调用它3次并相加即得问题的解。

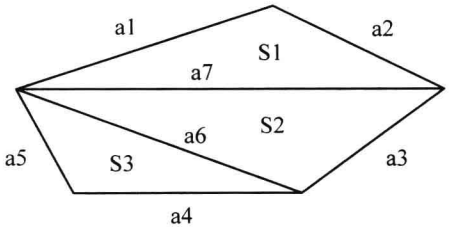

图1 多边形面积示意图

（2）算法设计：将算法框架中的一些关键算法细化，直到能用具体的程序设计语言表达为止。

① 输入a1，a2，a3，a4，a5，a6，a7；

② 计算s=ts（a1，a2，a7）+ts（a3，a6，a7）+ts（a4，a5，a6）；

③ 输出s。

（3）算法细化求精：对第①步算法求精：input（a1，a2，a3，a4，a5，a6，a7）；

对第②步算法求精：设三角形的三边长为a，b，c，则求三角形面积ts的算法如下：

第一步，计算 p=(a+b+c)/2;；

第二步，计算 ts=$\sqrt{p(p-a)(p-b)(p-c)}$ 。

（4）编写程序：这里我们用 C 语言进行编码。

四、课程知识结构

程序设计不仅要让学生掌握扎实的语法知识，而且应该重点培养学生的编程能力和创新思维。创新思维的培养是潜移默化的，教材应该在创新思维方面加以引导，培养学生发现问题、分析问题、解决问题的能力。为了构建知识、技能体系，实现理论与实践教学的完美结合，将“教、学、做”融为一体，全面强化学生能力素质的培养，课程教材由《C/C++程序设计》和《C/C++程序设计学习辅导》组成，并采用循序渐进的教学方式，即基础层→提高层→引申层，课程教学层次结构如图 2 所示。

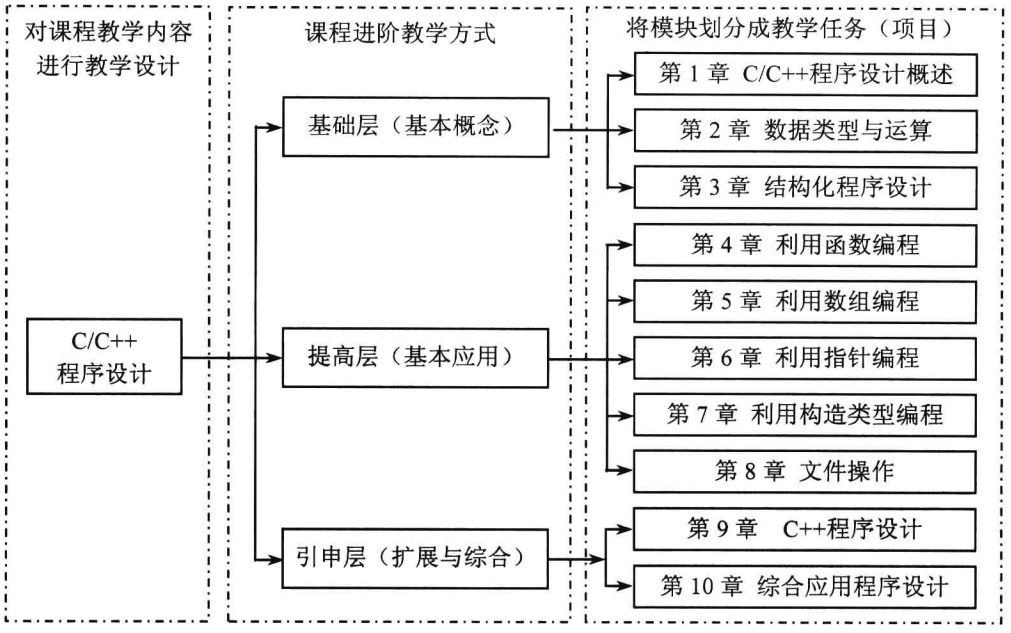

图 2 课程层次结构模块示意图

1.《C/C++程序设计》

《C/C++程序设计》是该课程教学的主教材，主教材各章的知识结构如图 3 所示。

图 3 主教材各章知识结构

（1）问题描述：描述本节所要解决的问题以及解决该问题的基本思路和基本方法。

（2）语法结构：根据所描述的问题，列出用 C/C++解决该问题时所涉及的基本语法结构。

（3）实例程序：根据语法结构编写实例程序，举一反三，融会贯通。

（4）算法案例：为了加强对知识点的理解和巩固，每章均以“算法案例程序设计”概括本章的教学内容，将语法结构与算法分析相结合。在第 2～10 章的“算法案例程序设计”中，

每章安排有 4 个典型案例，每个案例都以案例描述、案例分析、算法描述和程序实现四个步骤，全面概述本章的知识内容。所有的算法案例大都是具有理论研究价值、实际应用价值和颇有趣味性的经典算法，以此培养学生对程序设计的学习兴趣，提高学生实际编程能力。

（5）本章小结：指出本章的重点教学内容和知识点，以便于学生复习、总结和提高。

（6）本章习题：各章习题都设有四种题型：问答题、选择题、填空题、编程题，以此巩固所学知识内容。其中，选择题和填空题都是近年来全国 C 语言等级考试笔试真题，这样极有利于适应 C 语言等级考试中的笔试。

2.《C/C++程序设计学习辅导》

《C/C++程序设计学习辅导》由预备知识和知识技能两部分组成。前者为国家级 C 语言二级考试基础知识部分；后者与主教材内容对应，各章的知识结构如图 4 所示。

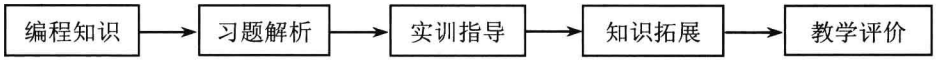

图 4　辅助材各章知识结构

（1）编程指导：包括两个方面的内容：“编程知识要点”和“常见编程错误”。其中：“编程知识要点”对应主教材中的知识要点，也是等级考试要点；“常见编程错误”给出学生在编程过程中容易出错的问题并予以解析，以进一步加深对基本概念的理解和语法规则的掌握，提高编写和调试程序的能力。

（2）习题解析：通过对问答题、选择题、填空题、编程题的剖析，进一步加深对各知识点的理解，并提高学生的应试能力。这里需要强调指出的是：在学习过程中，希望读者认真独立完成作业，“习题解析”只能作为学习参考。否则，便脱离了“辅导”的涵义。

（3）实训指导：C/C++语言是一门实践性很强的课程，为了强化动手能力的培养，对各章提出了实训要求。C/C++程序设计实训不仅包括语言本身，而且还包括 C/C++语言编程环境。我们要求学生能在 Turbo C++ 3.0 或 Visual C++ 6.0 环境下熟练编辑、编译、调试程序。

各章的实训内容包括：程序验证、程序填空、程序修改和程序设计。设计这些实训项目的目的是在提高学生动手能力的同时，引导学生对问题的思考，以培养学生的创新意识和创新能力。其中，程序填空、程序修改和程序设计都是近年来全国 C 语言等级考试机试真题。

（4）知识拓展：旨在使学生在掌握基本编程知识的基础上，向知识的深度和广度延伸，使学生对相关知识有一个更为全面的了解。与此同时，提高学生的学习兴趣，激发学生的钻研意识和创作热情，加速学生成长过程。

知识拓展以算法和数据结构为重点，将穷举算法、迭代算法、递推算法、递归算法、回溯算法、贪心算法、动态规划、运算模拟、排序算法、查找算法、线性表、队列、栈、树、图等知识合理安排在辅助教材中，与主教材中的“算法案例程序设计”相对应。这些算法知识为设计具有专业水准的应用程序，奠定了良好的理论基础。

（5）教学评价：为了检验教与学的效果，教学评价贯穿于课程始终。我们认为，科学的教学评价体系是实现课程目标的重要保障。教学评价包括两个方面：一是学生的自我评价，检验学生学习的三种能力（理解能力、技术能力、实践能力）。二是对教师教学能力和态度的评价，检验教师的教学方法、教学手段、教学经验和教学态度。因此，教学评价既是学生对掌握各章理论知识和实践动手能力的自我检测，也是对教师教学水平和态度的检测。通过对教与学的评价，不断改进“学”与“教”的方法，提高“学”与“教”的效果。

第 9 章介绍了 C++程序设计，其教学目的一是通过“C++对 C 语言的扩充”比较，使读者对 C++有初步的了解，消除学生认为 C++难学的恐惧心理，并为深入学习 C++打下基础；二是通过对象和类的学习，让学生了解面向对象程序设计方法的基本概念。

面向对象程序设计是当前程序设计的热门话题。面向对象程序设计方法主要是解决大型软件的设计问题。只有编写过大型程序的人才会体会到 C 的不足和 C++的优点。C++是一种大型语言，其功能、概念和语法规定都比较复杂，要深入掌握它需要花较多的时间，尤其是需要有较丰富的实践经验。正是这个原因，用 C++编程的主要是软件专业人员，学校里一般专业的程序设计课程任务主要是进行程序设计的基本训练。因此，我们认为当前对大多数学生来说，应先掌握好 C 语言程序设计，有了 C 语言的基础在需要时再学习 C++不会太困难。由于面向对象程序设计方法是在面向过程程序设计方法的基础上发展起来的，而面向对象程序设计语言 C++又是基于 C 语言的扩充，有了 C 语言基础，然后将 C++与 C 语言进行比较，这样，对初学者深入学习、理解和掌握 C++是极为有利的。

为了全面提高学生综合应用能力，第 10 章“综合应用程序设计”以工程项目的形式，将每一个项目设计成四个环节：系统设计目标、系统需求分析、系统总体设计、系统功能实现，以此提高学生运用所学知识开发应用软件的综合设计能力，引导学生养成良好的设计思路和编程习惯，丰富学生编写和调试大型程序的经验，使其能够编写出风格优美、可读性强、易于维护的程序代码。

五、课程学习建议

对于初学者来说，程序设计课程具有一定的挑战性，在学习过程中要注意总结学习方法。在教学过程中，我们发现许多理工科学习很好的同学，对程序设计课程也感“头疼和畏惧”，究其原因，主要是因为没有掌握正确的学习方法。那么，什么是程序设计课程的学习方法呢？

程序设计课程与其他课程相比，其区别主要表现在两个方面：其一，程序设计的基础逻辑思想是归纳逻辑，内容组织方式呈现网状特征，也就是说在学习前面的内容时，很可能会涉及到后面才会详细介绍的知识点；其二，程序设计是一门实践性很强的课程，任何一道程序最终都是需要计算机计算，并且需要进行反复调试才能得出正确结果。根据这样一些特点，提出几点建议，供学习者参考。

1. 采用整体学习方法

程序设计涉及的知识面很宽，而且很多知识点相互关联，是网状结构。所谓“整体学习方法”，就是在学习前面的内容时，如果遇到不懂的内容，先暂时接受，不要考虑为什么，继续学习后面的内容，注意各章各节的重点内容，整个内容学习结束后再回头解决前面的问题。这样反复学习，不断加深理解，每一次都会有新的收获。例如，在主教材第 1 章中介绍的程序设计方法、C 语言程序的实现；在辅助教材各章中介绍的常见编程错误、知识拓展等，它们对整个编程都有指导和警示作用，必须前、后、左、右反复学习。这里所谓的“左”、“右”，是指主教材与辅导教材两者配合，相互补充。

2. 培养基本思维模式

注意培养学习程序设计的基本思维模式。首先，掌握基本的思维过程，从固定思维模式入手。在程序设计中，要掌握数据的“输入→处理→输出”这样一个基本顺序，不能颠倒，这里输入是为处理做准备，而处理是为输出做准备；其次，正确理解和应用“自顶向下，逐步求精”的基本原则和思想方法。在程序设计过程中，需要沿着解析实际问题、构建数据类型、设

计问题算法，形成编程思路。编写的程序必须简单、正确、易懂。编程技巧应建立在良好的编程风格基础之上。大量阅读一些典型程序，是形成良好编程风格的主要途径。

3. 掌握基本设计方法

无论是自然语言或者是程序设计语言，语法规则都不难学，困难的是能够用语言正确地表达思想。正如我们为了学好英语，必须掌握一定的“习惯用法”才能很好地使用英语一样，程序设计也有一些固定的“习惯用语”，这就是基本编程模式和简单算法。例如，交换两个变量的值，多项数据求和，多项数据求最大值、最小值，多项数据按指定格式输出等，这样一些程序片段虽然称不上是算法，但是会经常用到，我们可以称其为编程摸式。另外，基本的算法，如最大公约数算法、排序算法和查找算法等也是在编写大型程序中会经常用到的算法。对于基本编程模式和基本算法，要求在学习过程中能牢牢记住，并且能够灵活地应用。

4. 把握数据组织方式

程序的核心是算法，算法的核心是数据处理，只有经过有效组织的数据才便于算法处理。因此，数据组织方式在程序设计中是非常重要的。在计算机科学中，数据的组织方式称为数据结构。C 语言中的简单数据组织方式是基本数据类型的变量，高级数据组织方式主要有数组、字符串、链表、结构体和数据文件等内容。在学习过程中，对不同的数据组织方式，要充分理解并掌握它们的存储特点。

5. 通过大量编程实践

程序设计是一门实践性很强的课程，只有经过大量的编写程序和调试程序，才能够掌握程序设计方法，才能编写出解决实际问题的程序。千万不要只注意“语言”，而忽视“设计”。要知道，语言只是手段，设计才是目的。

学会程序设计重在实践，它有两个方面的含义：其一，正如学习数学要有足够量的练习才能融会贯通一样，学习程序设计也需要足够的上机编程实践才能找到编程的感觉；其二，即使是有经验的程序员，除了极其简单的问题可以一次写出正确的程序外，对于大多数问题，初次写出的程序都会存在或多或少的问题，因此上机调试是学习程序设计过程中必须锻炼的一项基本技能。在程序员中流传着一个“2/8 定律”，即对于一个需要编程解决的问题，编写出初步程序所用时间占总时间的 20%。而编出的程序正确与否，都要经过上机调试，程序调试、排错、修改和正确性测试等要用到总时间的 80%。因此，不要以为掌握了程序的基本结构和基本语法就能学好这门课程，这种想法违背了课程特点和学习规律。

但是，不能因此而产生畏难情绪。每一位学习 C/C++程序设计的人都应从程序设计实践中有所收获，获得快乐。事实上，通过程序调试，既可以发现错误和排除错误，又可以进一步理解程序的行为，它既是一次再学习的过程，也是全面提高编程水平的过程。

6. 合理选择开发平台

作为 C/C++语言的初学者，建议选择那些操作简单、易学的开发平台来作为 C/C++语言程序设计的实践平台，这样可以将注意力更好地关注在 C/C++语言程序设计的方面，而不是更多地去了解开发平台本身。当学习者对 C/C++语言程序设计有了一定的认识和了解之后，可以根据程序设计的目的和要求的不同，选择不同的开发平台来完成 C/C++语言程序的开发。

目前，常用的 C/C++语言开发平台有 Turbo C++ 3.0 和 Visual C++ 6.0。考虑到各学校和教师的具体需求，在《C/C++程序设计学习辅导》中分别介绍了在 Turbo C++ 3.0 和 Visual C++ 6.0 这两种平台上编程和调试的方法。

7. 充分利用学习资源

充分利用网络获取学习资源也是非常重要的。互联网上有很多 C/C++语言程序设计的网站，是学习者很好的学习与交流平台。积极参与其中的活动，既可以获取学习资源，又可以培养团队精神。以下列出一些学习网址，供读者参考：

http://ww.csdn.net（中国程序员大本营）

http://msdn.Microsoft.com/zh-cn/default.aspx（微软简体中文 MSDN 主页）

http://www.ncre.cn（全国计算机等级考试官方网站）

hnp://www.bccn.net（编程中国）

hnp://www.google.cn（谷歌搜索引擎）

hnp://www.baidu.com（百度搜索引擎）

六、课程教学资源

为了提高教学效果，我们为“C/C++程序设计”课程建成立体式、多元化的教学资源。教学资源结构组成如图 5 所示。

该资源包括特色鲜明的文字教材、内容丰富的计算机辅助教学软件和功能完善的教学专用网站三大部分。其中：计算机辅助教学软件包含理论教学 PPT 系统和实训教学 PPT 系统。其中，理论教学 PPT 系统是为了进一步加深对基本概念的理解，以便对那些难以用语言和文字表述清楚的抽象概念，利用 PPT 的动画演示进行形象、准确、客观的描述。实训教学 PPT 系统是为了帮助学生自我训练和检测操作技能。

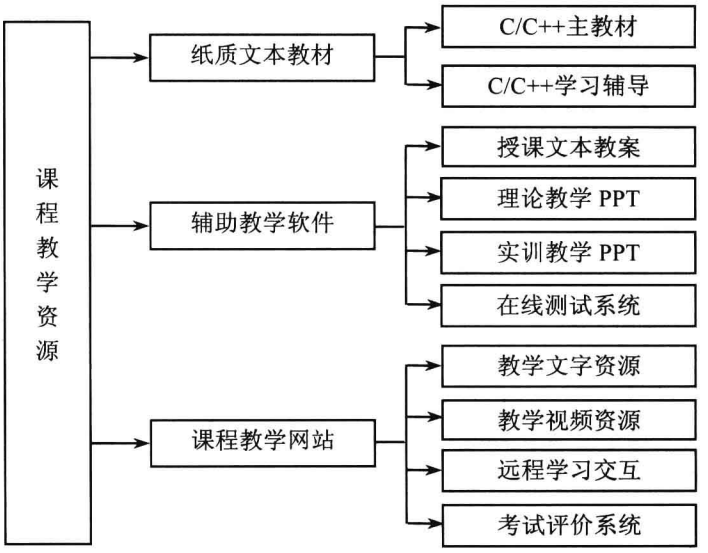

图 5 “C/C++程序设计”课程立体式教学资源的结构组成

此外，还有理论与实训教学的文本教案、上机考试与评价系统、教学专用网站等。教学网站提供了远程学习、讨论、练习、考试评价和资源下载等教学支持手段。

总之，无论是文字教材还是教学资源建设都力求有特色风格、有创新性、先进性和示范性，能有效地提高教学效果和教学质量。