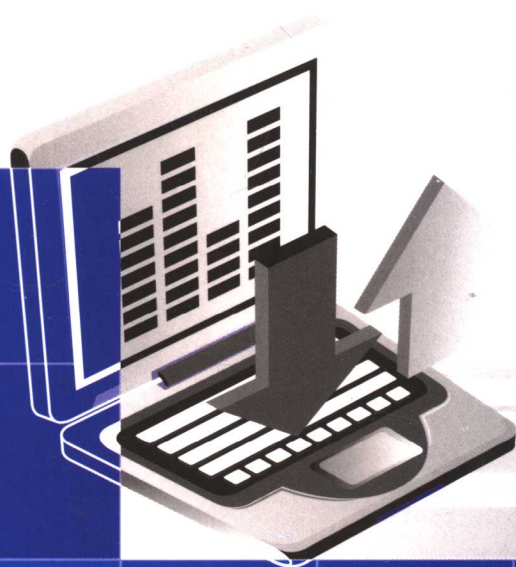


21st CENTURY
实用规划教材

21世纪全国应用型本科

计算机系列 实用规划教材



C++程序设计

主 编 刘天印 李福亮
副主编 张文国 李 兰



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国应用型本科计算机系列实用规划教材

C++程序设计

主 编	刘天印	李福亮
副主编	张文国	李 兰
参 编	代景霞	史玉珍



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

C++语言作为一门优秀语言,其严密的语法和强大的功能影响了随后出现的所有编程语言(如 Java 和 C#等)。

本书循序渐进地介绍了 C++程序设计的各个方面,主要内容有: C++语言与面向对象概述、基本数据类型及表达式、C++与结构化程序设计、数组、函数与预处理、指针、构造数据类型、类与对象、继承与派生、多态性、I/O 流与文件、模板与异常处理等。本书结构合理、应用性强,各章都有内容提要、学习目标、实例分析和归纳小结,示例简易而典型,课后还有丰富的编程练习。

本书由教学一线的资深教师编写,浅显易懂、图文并茂,适合作为各类高等院校 C++语言或面向对象程序设计等课程的教材,也可作为教师、学生或程序开发人员的参考书。

图书在版编目(CIP)数据

C++程序设计/刘天印,李福亮主编. —北京:北京大学出版社, 2006.1

(21 世纪全国应用型本科计算机系列实用规划教材)

ISBN 7-301-10466-9

I. C… II. ①刘… ②李… 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(2005)第 161307 号

书 名: C++程序设计

著作责任者: 刘天印 李福亮 主编

责任编辑: 李 虎 周 欢

标准书号: ISBN 7-301-10466-9/TP · 0873

出 版 者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://cbs.pku.edu.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667

电子信箱: pup_6@163.com

排 版 者: 北京东方人华北大彩印中心 电话: 62754190

印 刷 者: 北京宏伟双华印刷有限公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 25.25 印张 550 千字

2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷

定 价: 33.00 元

《21 世纪全国应用型本科计算机系列实用规划教材》

专家编审委员会

(按姓氏笔画排名)

主 任 刘瑞挺

副主任 胡昌振 段禅伦

崔广才 谢红薇

委 员 叶俊民 杨 璐 陈天煌 范冰冰

陈仲民 胡 明 秦 锋 龚声蓉

《21 世纪全国应用型本科计算机系列实用规划教材》

分系列专家编审委员会名单

(按姓氏笔画排名)

计算机应用技术——

主 任	胡昌振				
副主任	杨 璐	龚声蓉			
委 员	云 敏	马秀峰	李 明	肖淑芬	周松林
	杨长生	钟 声	赵忠孝	高 巍	

软件开发与软件工程——

主 任	谢红薇				
副主任	叶俊民	陈天煌			
委 员	王建国	孙 辉	吕海莲	李福亮	何朝阳
	张世明	陈佛敏	贺 华	赵绪辉	徐庆生
	徐 辉				

硬件与网络技术——

主 任	崔广才				
副主任	范冰冰	胡 明			
委 员	龙冬云	冯嘉礼	曲朝阳	汤 惟	张有谊
	董春游	程小辉			

专业基础课——

主 任	段禅伦				
副主任	陈仲民	秦 锋			
委 员	王昆仑	王 虹	仇 汶	田敬军	刘克成
	朴春慧	吴晓光	苏守宝	陈付贵	咎风彪
	谭水木	魏仕民			

信息技术的应用化教育

(代丛书序)

刘瑞挺/文

北京大学出版社第六事业部组编了一套《21 世纪全国应用型本科计算机系列实用规划教材》。为了做好这项工作，他们制订了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向应用、任务驱动、注重实例、培养能力；在风格上力求文字精练、脉络清晰、图表丰富、版式明快。

一、组编过程

2004 年 10 月，该部开始策划教材丛书，分别派出编辑深入各地高校，了解教学第一线的情况，物色合适的作者。2005 年 4 月 16 日在北京大学召开了“《21 世纪全国应用型本科计算机系列实用规划教材》研讨会”。来自全国 73 所院校的 102 位教师汇聚一堂，共同商讨应用型本科计算机系列教材建设的思路，并对规划选题进行了分工。2005 年 7 月 21 日在青岛又召开了“《21 世纪全国应用型本科计算机系列实用规划教材》审纲会”。编审委员会成员和 46 个选题的主编、参编，共 100 多位教师参加了会议。审稿会分专业基础课、软件开发与软件工程、硬件与网络技术、计算机应用技术等小组对大纲及部分稿件进行了审定。力争使这套规划教材成为切合当前教学需要的高质量的精品教材。

二、转变观念

为了搞好这套教材，要转变一些重要的观念。

首先，需要转变的观念就是大学及其培养人才的定位。大学并不都是“研究型”的，每个大学生不一定都当科学家。事实上，大多数学校应该是“应用型”的，大学生将直接进入社会基层、生产一线、服务前沿，成为各行各业的实践者和带头人。

其次，应该转变的观念就是教材建设的思路。许多人偏爱于“研究型”的教材，即使写“应用型”教材，也多半是前者的删繁就简、避虚就实，这样还不能产生真正“应用型”的教材。因此，以“学科”为中心、追求雄厚“理论基础”的传统应该被以“应用”为导向、追求熟练“实践技能”的思路所取代。

第三，必须转变对计算机技术的认识。20 年前，有人把计算机技术理解为 BASIC 编程；10 年前，有人把 Windows 95 和 Word 称为计算机文化；今天，中小学陆续开出《信息技术》课，有人对此怀疑观望，其实它意义深远。以计算机为核心的信息技术，今后 20 年的发展主题将是在各个领域的应用普及。大学计算机应用型本科的教材建设应该面向信息技术的深入应用，而不是相反，因为信息时代已经不是遥远的未来。

三、信息技能

以计算机为核心的信息技术，从一开始就与应用紧密结合。例如，ENIAC 用于弹道计算，ARPANET 用于资源共享以及核战争时的可靠通信。即使是非常抽象的图灵机模型，也

与第二次世界大战时图灵博士破译纳粹密码的工作相关。

今天的信息技术有三个重要的特点：

第一，信息技术是计算机与通信技术融合的辉煌成果。长期以来，计算机技术和通信技术并行不悖地独立发展。20 世纪后半叶，两者相互渗透，产生了程控电话、数据通信、网络技术、高清晰电视，世界各国构建了全球的、宽带的、网站密布的信息高速公路，出现了无处不在的手机通信和移动办公系统、随身听、数码摄录相机、家庭影院、智能控制系统，还有越来越多的嵌入式系统。人们的工作方式和生活方式都发生了质的飞跃。

第二，信息技术与各行各业紧密结合。我国的职业门类有：农林牧渔、交通运输、生化与制药、地矿与测绘、材料与能源、土建水利、制造、电气信息、环保与安全、轻纺与食品、财经、医药卫生、旅游、公共事业、文化教育、艺术设计传媒、公安、法律，这些门类都需要信息技术。

第三，在发展初期，以计算机为核心的信息技术是一项专门技术，只有专家才需要它、才能掌握它，在专家与平民之间有很深的“信息鸿沟”。今天，信息技术已经不再是只有专家才需要、才能掌握的专门技能，而是普通人都需要、也都能掌握的基本信息技能。但是，“信息鸿沟”也迁移到普通人之间。具有信息优势的学生能良性循环，强者更强。

有了这样广阔的应用信息背景，再造计算机应用型本科的课程体系就有了基础。

四、能力结构

关于应用型计算机人才的能力结构，我们不用“宫殿”模型，而用“雄鹰”模型。前者是建筑学模型，适合描述学科；后者是生物学模型，适合描述人才。“雄鹰”模型包括主体、两翼、头部、尾部等，它有可成长性。

首先，数据是信息技术的主体，数据技术是基本功。通常，数据包括文字、公式、表格、图形、图像、动画、声音、视频等等。因此，你不仅要会录入文章、绘制图表，还应该会采集音乐、编辑视频。大家面对的是多媒体数据，应该能收集它、整理它，数据经过整理就成为有用的信息。

其次，信息技术的两翼是数据库技术和网络技术。为了管理好、使用好数据，就必然用到数据库技术，数据库技术是一切信息管理的基石。为了分享数据和信息，就需要网络技术。有了上述数据主体技术和两个“翅膀”，你应该可以起飞了。

但是能飞多高，能飞多远，还应该编程技术、智能技术、安全技术的支持。这相当于头尾各部分的作用。编程将使大家的信息技能游刃有余。人工智能使你飞得更远，安全技术能使你飞得更稳。

有人可能会责难我们，难道大学本科生还需要学习办公软件的技能吗？他们认为这是让人“笑掉大牙”的事。其实，办公软件是最重要的提高生产效率的应用软件，很容易使用，但各人使用效率的高低则十分悬殊。我们设想，今后大学生在入学前先学会计算机的基本操作，我们再开一门高级办公技术的课，通过严格的行业及个人行为规范，对学生进行应用化训练，养成正确的职业习惯。将来工作时能提高效率、改善质量、降低成本。这决不是贻笑大方的事。

五、初步规划

应用型本科教材的规划是一个长期的战略任务，不是短期的战术行为。因此，目前的规划教材不可能一步到位，还会保留一些传统的基础课。例如，数字电路与逻辑设计、微机原理及接口技术、单片机原理及应用等。即使是纯硬件专业的学生，如何学这些传统硬件课都值得商榷，何况是作为公共基础课呢？

我们将分门别类逐步建设好应用型本科的重点课程和教材：

- (1) 基础类教材：信息技术导论，计算机应用基础，高级办公技术，数据与操作，密码与安全，实用数据结构，实用离散数学……；
- (2) 数据库类教材：数据库原理与应用，信息系统集成，数据采掘与知识发现……；
- (3) 网络类教材：计算机网络，因特网技术，网络管理与安全，网站与网页设计……；
- (4) 编程类教材：面向对象程序设计，C++程序设计，Java 程序设计……；
- (5) 提高类教材：软件工程原理及应用，人工智能原理及应用……。

新教材要体现教育观念的转变，系统地研究普通高校教学改革的需求，优先开发其中教学急需、改革方案明确、适用范围较广的教材。注重规划教材的科学性、实用性、易学性，尽量满足同类专业院校的需求。教材内容应处理好传统与现代的关系，补充新知识、新技术、新工艺、新成果。

我相信北京大学出版社在全国各地高校教师的积极支持下，精心设计，严格把关，一定能够建设一批符合应用型人才培养目标的、适应计算机应用型人才培养模式的系列精品教材，而且能建设一体化设计、多种媒体有机结合的立体化教材，为各门课程配套电子教案、学习指导、习题解答、课程设计等辅导资料。让我们共同努力吧！

刘瑞挺教授 曾任中国计算机学会教育培训委员会副主任、教育部理科计算机科学教学指导委员会委员、全国计算机等级考试委员会委员。目前担任的社会职务有：全国高等院校计算机基础教育研究会副会长、全国计算机应用技术证书考试委员会副主任、北京市计算机教育培训中心副理事长。

本系列教材编写目的和教学服务

《21 世纪全国应用型本科计算机系列实用规划教材》在全国的各位编写老师的共同辛勤努力下，在编委会主任刘瑞挺教授和其他编审委员会成员的悉心指导下，经过北京大学出版社第六事业部各位编辑的刻苦努力，终于与师生们见面了。

教材编写目的

目前，我国高等教育正迎来一个前所未有的发展机遇期。高等教育的发展已进入到一个新的阶段。高等本科院校也逐渐演变成“研究型、学术型”和“应用型、就业型”两类。

作为知识传承载体的教材，在高等院校的发展过程中起着至关重要的作用。但目前教材建设却远远滞后于应用型人才培养的步伐，许多院校一直沿用偏重于研究型的教材，应用型教材比较缺乏，这势必影响应用型人才的培养。

为顺应高等教育普及化迅速发展的趋势，配合高等院校的教学改革和教材建设，坚持“因材施教”的教学原则，注重理论联系实际，全面促进高等院校教材建设，进一步提高我国高校教材的质量，北京大学出版社大力推出高校“应用型本科”各专业相关教材。本系列教材不仅讲解基础理论技术，更突出工程实际应用，注重技术与应用的结合。

本套计算机系列教材的编写思想主要如下：

(1) 要符合学校、学科的计算机课程设置要求。以高等教育的培养目标为依据，注重教材的科学性、实用性、通用性，尽量满足同类专业院校的需求。

(2) 要定位明确。要准确定位教材在人才培养过程中的地位和作用，正确处理系列教材与系列课程、读者层次的关系，面向就业，突出应用。

(3) 合理选材和编排。教材内容应处理好传统内容与现代内容的关系，大力补充新知识、新技术、新工艺、新成果。根据教学内容、学时、教学大纲的要求，制定模块化编写体例，突出重点、难点。

(4) 体现建设“立体化”精品教材的宗旨。提倡为主干课程配套电子教案、学习指导、习题解答、课程设计、毕业设计等教学配套用书。

教学服务

1. 提供教学资源下载。本系列大部分教材中涉及到的实例（习题）的原始图片和其他素材或者是源代码、原始数据等文件，都可以在相关网站上下载。每本教材都配有 PPT 电子教案，老师可随时在网络上下载并可修改为适合自己教学的 PPT。

2. 提供多媒体课件和教师培训。针对某些重点课程，我们配套有相应的多媒体课件，对大批量使用本套教材的学校，我们会免费提供多媒体课件。另外，我们还将免费提供教师培训名额，不定期组织老师进行培训。

3. 欢迎互动。欢迎使用本系列教材的老师和同学提出意见和建议，有建议设的将给予奖励；同时有教材或者专著出版要求的老师，请与我们联系。

北京大学出版社第六事业部(<http://www.pup6.com>)

前 言

C++语言是从C语言继承发展而来的一种混合型的面向对象的程序设计语言。一方面C++语言全面兼容C语言，另一方面C++语言支持面向对象的方法。C++语言作为一门优秀语言，其严密的语法和强大的功能影响了随后出现的所有编程语言（如Java和C#等）。C++语言也随着时代的嬗变而在不断发展，自发行Visual Studio. NET后Visual C++ 6.0版升级为Visual C++ NET版，适应网络时代编程要求，使广大爱好C++语言的程序员振奋喜悦。

本书既可以作为程序设计的入门语言来学习，也可以作为掌握面向对象程序设计方法的进阶语言来学习。需要说明的是，学习C++语言不必首先学习C语言，因为C++本身就是一个完整的程序设计语言，并且兼容面向过程的C语言。本书在编写中兼顾到这两类读者的需求。

要学好程序设计，学生不仅需要掌握编程语言，还要掌握基本的数据结构和程序设计方法，加强应用型能力的培养，具有分析问题和解决问题的能力。本书在编写时始终围绕该目标，循序渐进地介绍了C++语言，全书结构合理、应用性强，各章都有内容提要、学习目标、实例分析和归纳小结，书中结合实例讲解基本概念和方法，课后习题类型多样、数量丰富。

本书共12章。第1章介绍了计算机语言的发展、C++程序的基本组成、面向对象概述和C++程序的开发过程。第2章介绍了C++语言的基本数据类型、运算符与表达式以及类型转换等。第3章介绍了结构化程序设计方法和C++继承C语言的结构化特征，即三大控制结构——顺序、选择和循环结构，以及转向控制语句break、continue和goto语句等。第4章讨论了数组，主要内容包括：一维数组的定义与使用、多维数组的定义与使用、字符串与字符数组。第5章讲述了函数和预处理的知识，函数是C++程序的基本组成单元，C++提供了三种预处理指令——宏、文件包含和条件编译。第6章介绍的是指针，读者可以了解指针与指针变量的概念，指针变量可以指向变量、数组、字符串，也可以指向函数甚至是另一个指针变量。第7章介绍了C++语言的构造数据类型，包括结构体、联合体、枚举类型、typedef声明等，并对链表作了简介。第8章讲述的是类与对象，围绕面向对象的三大特征：封装性、继承性和多态性，依次介绍了类和对象的定义方法、构造函数与析构函数、内联和重载在类的成员函数上的应用、静态成员、友元函数和友元类、子对象等的概念和使用，最后介绍了对象指针和对象数组。第9章讲述类的继承特性，包括：基类与派生类、单继承、多继承等，继承是面向对象程序设计的关键概念之一，有了继承，才使面向对象程序设计真正进入了实用。第10章讲述了类的另一个重要特性，即多态性。C++中的多态性有以下几种实现形式：函数重载、运算符重载、虚函数等，多态从实现的角度来讲可以分为两类：编译时的多态和运行时的多态，纯虚函数是在基类中说明的虚函数，带有纯虚函数的类是抽象类。第11章介绍了流的概念、流类库的结构和使用，C++语言中没有输入/输出语句，但C++编译系统带有一个面向对象的I/O软件包，它就是I/O流类库，它提供了一个功能完备的输入输出操作，流类库使用户能够很方便地对各种外设进行数据

的输入输出操作。第 12 章讨论的是模板与异常处理，包括：模板、异常处理的基本思想、C++异常处理的实现、异常处理中的构造与析构、标准程序库异常处理等。附录部分对 C++ 语言的编程进行了规范。

本书由刘天印和李福亮主编。第 1 章、第 7 章、第 10 章由河北经贸大学的李福亮编写，第 2 章、第 3 章及附录由黄石理工学院的刘天印编写，第 4 章由平顶山学院的代景霞编写，第 5 章、第 6 章、第 8 章由沈阳化工学院的张文国编写，第 9 章由平顶山学院的史玉珍编写，第 11 章、第 12 章由青岛理工大学的李兰编写，全书由刘天印统稿。另外感谢北京大学出版社提出了许多宝贵的意见。

本书由教学一线的资深教师编写，浅显易懂、图文并茂，适合作为各类高等院校 C++ 语言或面向对象程序设计等课程的教材，也可作为教师、学生或程序开发人员的参考书。另有配套教材：《C++程序设计实验指导与习题解答》，对本书中的习题进行了补充，并提供解答，与本书各章配套的实验，对于读者理解和熟练掌握书中相关内容提供帮助。我们还提供各章实例代码，可以从北京大学出版社网站下载。

感谢读者选择使用本书，欢迎广大教师、同行专家以及各位读者批评指正。

作 者

2005 年 9 月

目 录

第 1 章 C++语言与面向对象概述	1	2.3.2 关系运算符和关系表达式	37
1.1 计算机语言的发展概述	1	2.3.3 逻辑运算符和逻辑表达式	38
1.1.1 计算机程序设计语言的发展	1	2.3.4 位运算符	39
1.1.2 程序设计方法的发展	2	2.3.5 赋值运算符和赋值表达式	42
1.1.3 C++的特点	4	2.3.6 条件运算符	43
1.1.4 程序的演示	5	2.3.7 逗号运算符	43
1.2 C++程序的基本组成	6	2.3.8 sizeof 运算符	43
1.2.1 C++的基本词法和规范	6	2.3.9 运算符的优先级与结合性	44
1.2.2 简单的 C++程序	7	2.4 类型转换	45
1.2.3 C++程序结构说明	8	2.4.1 自动类型转换	45
1.3 面向对象概述	9	2.4.2 强制类型转换	46
1.3.1 面向对象的基本概念	9	2.5 实例分析	47
1.3.2 面向对象的基本特征	10	2.6 本章小结	48
1.3.3 面向对象的软件开发	12	2.7 习题二	49
1.4 C++程序的开发	13	第 3 章 C++与结构化程序设计	52
1.4.1 C++程序的开发	13	3.1 C++基本语句	52
1.4.2 Visual C++ 6.0 集成开发		3.2 结构化程序设计方法	53
环境简介	13	3.2.1 模块化设计	53
1.4.3 开发环境的初步实践	15	3.2.2 结构化编码	54
1.5 本章小结	19	3.2.3 结构化程序设计方法的	
1.6 习题一	19	特点	54
第 2 章 基本数据类型及表达式	22	3.3 顺序结构程序设计	55
2.1 基本数据类型	23	3.4 选择结构程序设计	55
2.1.1 整型	23	3.4.1 if 语句	55
2.1.2 字符型	24	3.4.2 if 语句的嵌套	60
2.1.3 浮点型	26	3.4.3 switch 开关语句	60
2.1.4 布尔型	28	3.5 循环结构的实现	62
2.1.5 类型修饰符	29	3.5.1 while 语句	62
2.2 常量与变量	30	3.5.2 do-while 语句	63
2.2.1 常量	30	3.5.3 for 语句	64
2.2.2 变量	33	3.5.4 循环的嵌套	67
2.3 运算符与表达式	35	3.5.5 循环与选择的相互嵌套	68
2.3.1 算术运算符和算术表达式	35	3.6 转向控制语句	69

3.6.1 break 语句.....	69	5.5 作用域与生命期.....	128
3.6.2 continue 语句.....	71	5.5.1 作用域.....	128
3.6.3 goto 语句.....	72	5.5.2 全局变量和局部变量.....	130
3.7 实例分析.....	73	5.5.3 生命期.....	132
3.8 本章小结.....	79	5.6 编译预处理.....	135
3.9 习题三.....	80	5.6.1 宏定义(#define).....	136
第4章 数组.....	85	5.6.2 文件包含.....	138
4.1 一维数组.....	85	5.6.3 条件编译.....	140
4.1.1 一维数组的定义.....	85	5.7 实例分析.....	141
4.1.2 一维数组的初始化.....	86	5.8 本章小结.....	143
4.1.3 对一维数组元素的访问.....	87	5.9 习题五.....	144
4.2 多维数组.....	90	第6章 指针.....	148
4.2.1 二维数组的定义.....	91	6.1 地址和指针.....	148
4.2.2 二维数组的初始化.....	91	6.2 变量的指针和指向变量的 指针变量.....	149
4.2.3 对二维数组元素的访问.....	92	6.2.1 指针变量的定义.....	149
4.3 字符数组.....	95	6.2.2 指针变量的引用.....	151
4.3.1 字符数组的定义.....	95	6.2.3 指针变量的运算.....	152
4.3.2 字符数组的初始化.....	95	6.2.4 指针变量作为函数参数.....	154
4.3.3 字符数组的引用.....	96	6.3 指针与数组.....	155
4.3.4 字符串.....	97	6.3.1 指向数组元素的指针.....	155
4.3.5 字符串处理函数.....	100	6.3.2 通过指针引用数组元素.....	156
4.4 实例分析.....	103	6.3.3 数组名作函数参数.....	159
4.5 本章小结.....	111	6.4 指针与字符串.....	162
4.6 习题四.....	111	6.4.1 字符串的表示形式.....	162
第5章 函数与预处理.....	115	6.4.2 字符串指针作函数参数.....	163
5.1 函数的定义.....	115	6.5 指向函数的指针.....	164
5.1.1 函数的定义格式.....	115	6.5.1 用函数指针变量调用函数.....	164
5.1.2 函数的原型声明.....	116	6.5.2 用指向函数的指针作函数 参数.....	165
5.2 函数的调用和参数.....	117	6.6 指针数组和指向指针的指针.....	166
5.2.1 函数的调用.....	117	6.6.1 指针数组的概念.....	166
5.2.2 函数的参数.....	118	6.6.2 指向指针的指针.....	167
5.3 函数的嵌套和递归调用.....	122	6.6.3 指针数组作 main 函数的 形参.....	169
5.3.1 函数的嵌套调用.....	122	6.7 实例分析.....	170
5.3.2 函数的递归调用.....	123	6.8 本章小结.....	173
5.4 内联函数和重载函数.....	125	6.9 习题六.....	173
5.4.1 内联函数.....	125		
5.4.2 函数的重载.....	126		

第 7 章 构造数据类型	177	8.6.1 友元函数	230
7.1 结构体	177	8.6.2 友元类	231
7.1.1 结构体类型的定义	178	8.7 嵌套类和子对象	233
7.1.2 结构体变量的定义与 初始化	179	8.7.1 嵌套类	233
7.1.3 结构体成员的访问	182	8.7.2 前向引用	234
7.1.4 结构与函数	187	8.7.3 子对象	235
7.1.5 结构与链表	193	8.8 对象指针和对象数组	237
7.2 联合体	198	8.8.1 指向对象的指针	237
7.2.1 联合体的定义	198	8.8.2 this 指针	238
7.2.2 访问联合体的成员	199	8.8.3 对象数组	240
7.3 枚举类型	201	8.9 常对象和常成员	241
7.3.1 枚举类型及枚举变量的 定义	201	8.9.1 常对象	241
7.3.2 枚举元素的引用	202	8.9.2 常成员	243
7.4 typedef 声明	203	8.9.3 常类型的函数参数传递	245
7.5 实例分析	203	8.10 实例分析	247
7.6 本章小结	206	8.11 本章小结	250
7.7 习题七	207	8.12 习题八	250
第 8 章 类与对象	212	第 9 章 继承与派生	255
8.1 类和对象的定义	212	9.1 基类与派生类	256
8.1.1 类的定义	212	9.1.1 基本术语	256
8.1.2 类成员的访问控制	214	9.1.2 派生类的定义格式	256
8.1.3 对象的定义	215	9.1.3 派生类的继承方式	258
8.1.4 对象成员的访问方式	215	9.2 单继承	262
8.2 构造函数	216	9.2.1 基类成员的访问权限	262
8.2.1 构造函数的定义和分类	216	9.2.2 构造函数和析构函数	265
8.2.2 默认构造函数	218	9.2.3 作用域运算符“::”	270
8.2.3 复制构造函数	219	9.2.4 子类型和类型适应	270
8.2.4 类型转换构造函数	221	9.3 多继承	273
8.3 析构函数	222	9.3.1 多继承的概念	273
8.4 内联和重载	223	9.3.2 多继承的二义性	277
8.4.1 内联成员函数	223	9.3.3 虚基类的定义	279
8.4.2 重载成员函数	224	9.3.4 虚基类及其派生类构造 函数	281
8.5 静态成员	226	9.3.5 虚基类的应用	283
8.5.1 静态数据成员	226	9.4 实例分析	285
8.5.2 静态成员函数	228	9.5 本章小结	289
8.6 友元	230	9.6 习题九	290

第 10 章 多态性	294	11.3.2 浮点数精度和域宽的 使用.....	336
10.1 静态联编与动态联编.....	294	11.4 文件的操作.....	337
10.1.1 静态联编.....	294	11.4.1 文件流.....	337
10.1.2 动态联编.....	297	11.4.2 文件的打开方式.....	338
10.2 运算符重载.....	299	11.4.3 文本文件与二进制文件的 读/写.....	340
10.2.1 运算符重载的规则.....	300	11.4.4 顺序访问文件.....	343
10.2.2 运算符重载为成员函数.....	301	11.4.5 随机访问文件.....	344
10.2.3 运算符重载为友元函数.....	304	11.5 实例分析.....	346
10.3 虚函数.....	307	11.6 本章小结.....	348
10.3.1 虚函数的定义和使用.....	307	11.7 习题十一.....	349
10.3.2 虚析构造函数.....	309	第 12 章 模板与异常处理	355
10.4 纯虚函数与抽象类.....	309	12.1 模板.....	355
10.4.1 纯虚函数.....	309	12.1.1 函数模板.....	355
10.4.2 抽象类.....	310	12.1.2 类模板.....	360
10.4.3 动态联编的实现机理.....	311	12.2 异常处理的基本思想.....	363
10.5 实例分析.....	312	12.3 C++异常处理的实现.....	364
10.5.1 问题提出.....	312	12.3.1 异常处理的语法.....	364
10.5.2 类设计.....	312	12.3.2 异常接口声明.....	367
10.5.3 程序代码设计.....	313	12.4 异常处理中的构造与析构.....	367
10.6 本章小结.....	317	12.5 标准程序库异常处理.....	370
10.7 习题十.....	318	12.6 实例分析.....	371
第 11 章 I/O 流与文件	322	12.7 本章小结.....	374
11.1 I/O 流的概念及流类库结构.....	322	12.8 习题十二.....	374
11.1.1 C++流类的体系结构.....	322	附录 C++语言编程规范	379
11.1.2 预定义流对象.....	325	一、应用范围.....	379
11.1.3 标准 I/O 流类.....	326	二、设计模块和函数.....	379
11.1.4 提取运算符“>>”和插入 运算符“<<”.....	328	三、命名约定.....	381
11.2 用 ios 类的成员函数实现格式化 输入/输出.....	328	四、使用常量和枚举值.....	382
11.2.1 ios 类的流成员函数.....	328	五、变量.....	383
11.2.2 左右对齐.....	331	六、代码的格式化.....	384
11.2.3 设置填充字符.....	332	七、代码的注释.....	385
11.2.4 设置及清除格式标志.....	333	八、控制代码流.....	387
11.2.5 流错误状态.....	334	参考文献	388
11.3 用操作符实现格式化输入/输出.....	335		
11.3.1 整数的格式化输入/输出.....	335		

第 1 章 C++语言与面向对象概述

教学提示：C++是在 C 语言的基础上拓展而来的。C++引入了面向对象的机制，同时又充分保留了 C 语言的简洁性和高效性，并且与 C 语言完全兼容，具有 C 语言高效灵活、功能强大、可移植性好等诸多优点，是面向对象程序设计(Object-Oriented Programming)的最佳语言之一。

教学要求：学习 C++，一是掌握面向过程程序设计语言的基本方法和技能，二是以 C++为手段，通过编程训练可以更具体地理解和掌握面向对象的一些基本特征(如抽象、封装、继承、多态和动态联编等)，学习和掌握面向对象程序设计的基本思想和技术，为学习类似的其他语言(如 Java 语言、Visual C++、Visual J++、C#等)打下良好的基础。

学习目标：

- 了解 C++的主要特点；
- 理解和掌握 C++程序的基本结构、C++的基本词法和规范；
- 了解面向对象技术的基本概念；
- 初步掌握 Visual C++集成开发系统的步骤和方法；
- 通过实例演示，激发学习兴趣，明确 C++学习的目标。

1.1 计算机语言的发展概述

1.1.1 计算机程序设计语言的发展

计算机的工作体现为顺序执行程序，程序是控制计算机完成特定功能的一组有序指令的集合，编写程序所使用的语言称为程序设计语言，它是人与计算机之间进行信息交流的工具。计算机程序设计语言是计算机可以识别的语言，用以描述解决问题的方法，供计算机阅读和执行，与一般语言一样，它具有一套语法、词法规则的系统。从 1946 年世界上第一台计算机诞生以来，短短的 50 年间计算机科学得到了迅猛发展，程序设计语言的发展从低级到高级，经历了机器语言、汇编语言、高级语言到面向对象语言的多个阶段，具体过程如下所述。

1. 机器语言

计算机的硬件系统可以直接识别和执行的二进制指令(机器指令)的集合称为该种计算机的机器语言。早期的计算机程序是直接使用机器语言编写的，这种语言便于计算机直接识别，但编写出的程序难以理解和记忆。

2. 汇编语言

汇编语言将机器指令映射为一些可以被人读懂的助记符，如 ADD、MOV 等，它实际

上是与机器语言相对应的语言。由于计算机只能执行机器指令，故汇编语言需要编译后才能被识别，这一过程称为汇编。同机器语言一样，与人类的自然语言仍然相差甚远。

3. 高级语言

20 世纪 60 年代，出现了高级语言。**高级语言屏蔽了机器的细节，提高了语言的抽象层次，程序中可以采用具有一定含义的数据命名和容易理解的执行语句，接近人类的自然语言。**使用高级语言编程，一般不必了解计算机的指令系统和硬件结构，只需掌握解题方法和高级语言的语法规则，就可以编写程序。高级语言在程序设计时着眼于问题域中的过程，是一种面向过程的语言。这使得在书写程序时可以联系到程序所描述的具体事物，使计算机的编程语言前进了一大步。

这个时期，随着计算机的应用日益广泛地渗透到各学科和技术领域，也发展了一系列不同风格的程序设计语言，较为著名的有 FORTRAN、COBOL、BASIC、PASCAL、Turbo C 等。

4. 面向对象的语言

20 世纪 80 年代，出现了面向对象的编程语言。**面向对象的编程语言与以往各种编程语言的根本不同点在于：它设计的出发点就是为了能更直接地描述客观世界中存在的事物(即对象)以及它们之间的关系。**面向对象语言的出现改变了编程者的思维方式，使设计程序的出发点由问题域中的过程转向问题域中的对象及其相互关系，这种转变更加符合人们对客观事物的认识。因此，面向对象的方法更接近于自然语言，是人们对于客观事物更高层次的抽象。

面向对象的语言已形成几大类别：一类是纯面向对象的语言，如 SmallTalk 和 Eiffel；一类是混合型的面向对象语言，如 C++ 和 Objective C；还有一类是与人工智能语言结合形成的，如 LOOPS、Flavors、CLOS 以及适合网络应用的 Java 语言等。

用源语言书写的程序，称为**源程序**。源语言可以是汇编语言，也可以是高级程序设计语言；源程序需要翻译成机器语言的**目标程序**才能执行。

翻译程序有 3 种不同的类型：汇编程序、编译程序、解释程序。**汇编程序**完成把汇编语言写成的程序翻译成目标程序；**编译程序**将高级语言程序翻译成目标程序；**解释程序**也是将高级语言程序翻译成机器指令，它与编译程序的不同点在于它是边翻译边执行的。

1.1.2 程序设计方法的发展

计算机程序设计方法的发展大致经历了面向机器的方法、面向过程的方法和面向对象的方法等几个阶段。

1. 面向机器的方法

面向机器的方法使用机器语言或汇编语言。设计一个程序，基本上是这样一个过程：写出算法，验证算法，转换成机器语言，然后调试代码，直到产生希望的结果。

这种方法基本上是“软件作坊”式的，与人类的自然语言之间存在着巨大的鸿沟。在实现上程序员需要考虑大量的机器细节，软件开发的难度大、周期长，开发出的软件功能却很简单，界面也不友好。这种局面严重影响了计算机的普及与应用。