普通高等教育“十一五”国家级规划教材
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI

C++ CHENGXU SHEJI JICHU

C++程序设计基础

佟勇臣　边奠英　编著

中国电力出版社
http://jc.cepp.com.cn

普通高等教育“十一五”国家级规划教材
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI

C++ CHENGXU SHEJI JICHU

C++程序设计基础

编著　佟勇臣　边奠英
主审　叶乃文　商建云

中国电力出版社
http://jc.cepp.com.cn

内 容 提 要

本书为普通高等教育“十一五”国家级规划教材。

作为C++语言的入门教材，本书用简明的语言阐述了C++语言的概念，使不容易理解的概念变得通俗易懂，便于初学者学习。全书分三个部分。第一部分讲述C++语言基础。第二部分阐述面向对象的程序设计，这两部分都配有例题和练习。第三部分是上机指导与习题解答，上机指导给出了上机实验的详细步骤和过程，思考题与习题给出了详细的解答。书中使用了大量的例题和实验，对C++语言的应用进行讲解，使读者能深入了解C++语言的使用方法；每章之后都有思考题和练习题，使读者通过思考和练习巩固所学的知识。

本书可作为普通高等院校C++程序设计课程的教材，也可作为高职高专学生的教材或参考书，还可供初学C++语言的读者参考。

图书在版编目（CIP）数据

C++程序设计基础 / 佟勇臣，边奠英编著. —北京：中国电力出版社，2009

普通高等教育“十一五”国家级规划教材

ISBN 978-7-5083-9337-7

Ⅰ. C… Ⅱ. ①佟…②边… 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（2009）第147068号

中国电力出版社出版、发行

（北京三里河路6号 100044 http://jc.cepp.com.cn）

北京市同江印刷厂印刷

各地新华书店经售

*

2009年9月第一版 2009年9月北京第一次印刷

787毫米×1092毫米 16开本 28印张 684千字

印数 0001—3000册 定价 **45.00** 元

前　言

本书是作者多年从事 C++语言教学经验的结晶，是作者在 10 余年教授 C++语言课程的授课讲义的基础上，结合当前本科学生学习的特点进行修改补充而成。本书的特点如下。

（1）言简意明，通俗易懂，概念阐述明确，突出重点，重点、难点着重论述。满足本科学生学习程序设计语言的教学需要，着重强调应用能力的培养。

（2）尊重认识规律，内容的安排循序渐进，深入浅出，并以具体实例和实际应用来分析和阐述 C++语言中的概念和原理，尽量避免抽象的理论讲解，由感性到理性地安排和组织内容，便于学生掌握和运用。

（3）书中使用了大量的例题、习题和实验，内容丰富，通过这些内容的合理使用，结合习题和实验，使学生能深入了解 C++语言的使用方法，在较短的时间内掌握 C++语言。

（4）每章之后都有思考题和练习题，使学生通过思考和练习巩固所学的知识。对相同的题，书中在实验和习题中给出了不同的编程方法，目的是使学生能体会编程方法的多样性。

（5）为了对初学者负责，书中的例题、实验程序和习题的解答，都在 Visual C++ 6.0 环境下进行了上机调试，保证运行通过。

（6）采用流行的内容和理念，引导学生掌握最新的先进技术与成果，激发学生热情和兴趣，使学生深入学习相关知识，掌握和应用相关技术。

在本书的编写过程中，特别考虑到没有任何程序设计语言学习经历的初学者的学习，尤其是本科、大专和高职的学生，在初次学习 C++语言时的特点，适合他们使用。

全书分三个部分。第一部分讲述 C++语言的基础，用通俗易懂的语言阐述 C++的基本内容；第二部分阐述面向对象的程序设计，用简明的语言论述了面向对象设计的要点和方法。这两部分都有例题和练习与之配合，各重点部分除了讲解详细之外还用例题讲解其使用方法。第三部分是上机指导和思考题、习题的解答，上机指导给出了上机实验的详细步骤和过程，思考题与习题给出了详细的解答。本书虽然是针对本科层次的学生编写，由于有大量的例题和实验，以及翔实的习题解答，也可以作为大专、高职学生的教材或参考书。

本书的第 1～10 章、第 13～14 章由佟勇臣编写，第 12 章由边奠英编写，第 11 章由尹丽华编写。本书由佟勇臣统稿，叶乃文和商建云主审。

作　者

2009 年 6 月

目　录

第二篇 面向对象的程序设计

第三篇 上机指导与习题解答

第一篇 C++ 语 言 基 础

在本篇中将着重介绍面向对象的程序设计语言——C++语言的基本概念和基本方法。由于C++语言过于抽象，对没有任何程序设计基础的初学者来说不容易理解，所以，本篇将C++语言的基本概念和方法抽取出来进行介绍，使初学者能够容易理解和接受。以结构化程序设计方法为代表的面向过程程序设计是面向对象程序设计的基础，初学者只有掌握了基础，才能够深入学习面向对象程序设计的精华。

面向过程的程序设计方法是具体的（如C语言），编程的针对性强，因而易于理解和接受，但是，在大型、复杂的系统设计上，面向过程的方法显得力不从心，不如面向对象的方法得心应手，安全可靠。正是鉴于此种情况，本篇以初学C++程序设计语言的学生为对象，目的是使他们由此入门，能够学会和掌握功能强大的C++程序设计的技术和方法。

第1章 C++ 语 言 概 述

知识点

（1）C++语言的类与对象。
（2）C++语言类的封装性与继承性。
（3）C++语言类的抽象性与多态性。
（4）面向对象的程序设计。

难 点

（1）C++语言类的应用。
（2）面向对象的程序设计思想。

要 求

（1）熟练掌握以下内容：
① C++语言类与对象的运用。
② C++语言的关键字。
③ C++语言的语法规则与书写格式。
④ C++语言的编程特点。
（2）了解以下内容：
① C++语言的发展概况。
② C++语言的程序组成。

1.1 C++语言发展概述

C++语言发展至今，已成为版本众多、应用范围广泛的程序设计语言。C++语言比C语

言应用范围广，而 C 语言的概念、方法相对简单一些，比 C++语言更容易理解，这是因为 C++语言是面向对象的程序设计语言，而 C 语言是面向过程的。C++语言兼容 C 语言，有 C 语言基础的人更能够体会 C++语言功能的强大和程序设计的方便。

1.1.1 C++语言产生的背景

1. 从机器语言到汇编语言

自从 1946 年世界上第一台数字电子计算机 ENIAC 诞生以来，在这 60 多年间，计算机科学得到了迅猛发展，计算机及其应用已渗透到社会的各个领域，有力地推动了整个社会信息化的发展，计算机已成为信息化社会中必不可少的工具。

计算机之所以具有如此强大的功能，不仅因为它具有强大的硬件系统，而且依赖于软件系统。软件包括使计算机运行所需的各种程序及其有关的文档资料。计算机的工作是用程序来控制的，离开了程序，计算机将一事无成。程序是指令的集合，软件工程师将解决问题的方法、步骤编写成由指令序列组成的程序，输入到计算机的存储设备中，计算机执行这些指令序列，便可完成预定的任务。

所谓指令，就是计算机可以识别的命令。在人类社会中，有丰富的语言来表达思想、交流感情、记录信息，但计算机却不能识别它们。计算机所能识别的指令形式，只能是简单的“0”和“1”的组合。一台计算机硬件系统能够识别的所有指令的集合，称为它的指令系统。

计算机硬件系统可以识别的二进制指令组成的语言称为机器语言，计算机能够识别、执行，对于人类来说却是晦涩难懂，难以记忆。在计算机发展的初期，软件工程师只能用机器语言编写程序。在这一阶段，存在着人类的自然语言与计算机语言之间巨大的鸿沟，软件开发的难度大、周期长，开发出的软件功能简单。

后来，出现了汇编语言，它将机器指令映射为一些可以被人读懂的助记符，如 ADD、SUB 等。此时编程语言与人类自然语言之间的鸿沟有了缩小，但仍与人类的思维的方式相差甚远，程序员需要考虑大量的机器细节。尽管如此，从机器语言到汇编语言是一大进步，这意味着人与计算机系统不必使用同一种语言。程序员可以使用较适合人类思维习惯的语言，而计算机硬件系统仍只识别机器指令。这两种语言之间的沟通需要一种翻译软件来完成。汇编语言的翻译软件称为汇编程序，可以将程序员写的助记符直接转换为机器指令，由计算机识别和执行。

2. 从汇编语言到高级语言

汇编语言的出现虽然减轻了编程人员的工作量，但是，还要程序员考虑大量的机器细节，能否不考虑计算机的硬件细节而只管编程呢？高级语言的出现使这种想法得以实现。高级语言屏蔽了计算机硬件的细节，提高了语言的抽象能力，使程序员在编写程序时只考虑程序所描述的具体事物，是计算机编程语言的一大进步。

从 20 世纪 60 年代末，开始出现了结构化编程语言，这进一步提高了编程语言的层次。结构化数据、结构化语句、过程抽象等概念使程序更便于体现客观事物的结构和逻辑含义，使得编程语言与人类的自然语言更加接近。这两者之间的主要差距是程序中的数据和操作相分离，不能有效地组成与自然界中某一具体事物紧密对应的程序段。应用比较广泛的几种高级语言有 BASIC、C、Pascal 等语言。

3. 从面向过程的语言到面向对象的语言

面向对象的程序语言与面向过程的程序语言有着根本的不同，前者的出发点是为了更直

接地描述客观世界中存在的事物以及它们之间的关系。

编写程序是为了解决实际问题，这些问题所涉及的业务范围称为该程序的问题域。面向对象的编程语言是将客观事物看作具有属性和行为（或服务）的对象，通过抽象找出同一类事物的共同属性（静态特征）和行为（动态特征）。通过类的继承与多态可以实现代码重用，缩短软件开发的工作量和周期，使软件的风格统一。因此，面向对象的语言能够使程序直接反映问题域的本来面目，使软件开发人员能够用人类认识事物的思维方法进行软件的开发。

面向对象的程序设计语言经历了一个很长的发展阶段，很多语言或多或少地都引入了面向对象的概念，例如，Lisp语言、Simula67语言、Smalltalk语言以及Ada、Modula-2语言等，其中Smalltalk是第一个真正的面向对象的程序语言。

然而，应用最广的面向对象的程序设计语言是在C语言基础上发展起来的C++语言，由于C++对C兼容，而C语言又早已被广大程序员所熟知，所以，C++语言也就理所当然地成为应用最广的面向对象的程序设计语言。

4. 程序设计语言在危机中不断发展

程序设计语言随着软件技术的发展而快速发展，反过来它又推进了软件产业。20世纪60年代初软件界的主导语言是FORTRAN、COBOL和Algol 6.0，由于它们的数据类型单调，程序控制过多地依赖程序员的技巧，滥用GOTO语句等，使程序难读、难改、不易移植、可靠性差，形成了第一次软件危机。为了规范软件内部的混乱，20世纪70年代兴起了结构化的模块设计方法，Pascal、C语言相继出现，成为流行语言。由于使用了结构化的设计方法，大大提高了软件的质量和可靠性，使程序规模可达400余万语句，例如，美国导弹预警系统。

随着硬件的不断发展，人们希望编写规模更大的软件，这就使已经缓解的软件危机大有加剧之势。造成这种情形的主要原因是软件人员希望只要硬件能满足需要，就会编制更大的软件系统。软件系统大到一定的规模，现有的方法就无法管理与维护。结构化的程序设计方法在更大型的软件系统面前显得力不从心，出现的问题层出不穷；因此，人们又在探询新的软件开发方法。

20世纪70年代中期，有人将工程管理的方法应用到软件行业，兴起了软件工程。软件工程方法除了拥有成套的管理方法之外，在技术上强调模块性、抽象性、易维护、可修改和可移植等。20世纪80年代受益于软件工程方法，使软件系统的规模得以继续扩大，最大的软件系统已达4000余万句，例如，美国航天飞机监控系统。

20世纪90年代初，传统的软件工程方法对软件危机的缓解已不起作用，大型软件投资的失败，系统软件的不可靠，不易维护、不可移植等现象又大量存在。寻找新的软件系统的开发和表达方法，仍然是解决软件危机的非常重要的事情。在抽象数据类型和交互式环境设计思想的基础上，面向对象的程序设计方法的出现，给处于危机中的软件产业带来了希望。

面向对象技术，以其封装的模块性、类和实例的抽象性、继承机制提供的可扩充性等，无疑成为当代最成功的软件技术。软件产业的各个主导语言都纷纷向面向对象技术扩充，以求在竞争中立于不败之地，C++就是在这样的背景下从C语言演变而来。

到了20世纪90年代末，又出现了“可视化”编程技术，大大简化了编程的手段和方法，减少了编程的工作量，Visual C++就是这个时期出现的程序设计方法。Visual C++是将C++的编程方法进行了可视化处理，形成了独特的可视化编程技术，小到应用系统，大到开发工具，

它几乎可以设计一切程序软件。它不仅全面贯彻了面向对象的技术，而且在编译优化技术等方面，较其他同类产品也具有明显的优势。

1.1.2 C++语言的发展

开发C++语言的宗旨是使面向对象程序设计技术和数据抽象成为软件开发的一种真正的实用技术。所以C++语言的形成是一个发展和完善的过程。其研制和开发过程是以编译系统的有效实现为前提的，这正是C++语言能够成功的原因。

1. C++语言的诞生

1983年底，贝尔实验室的项目负责人决定把C++语言作为关键项目进行开发，项目负责人对设计者提出的第一个要求就是“C++语言要与C语言100%的兼容”。这个要求决定了C语言是C++语言的子集。C++语言自诞生起很快受到广泛的欢迎，与这个决定是分不开的，C++语言的用户数，从1985年的500，增至1991年的几十万。例如，Borland公司到1991年10月就售出C++编译器达50万套。

2. 由C语言到C++语言

C语言是C++语言的一个真子集，即C∈C++；C++语言是C语言的超集，这就是C语言与C++语言的关系。C++语言是C语言向面向对象的扩充，这种扩充持续了很多年。

（1）在1985年公布的C++ 1.0版之前，C++语言是“C with Class”阶段，此时，在C语言的基础上添加的特征主要有：①类及派生类；②公有和私有成员的区分；③类的构造函数和析构函数；④友元；⑤内联函数；⑥赋值运算符的重载等。

（2）在1985年公布的C++ 1.0版时，又增加了一些重要特征：①虚函数的概念；②函数和运算符的重载；③引用；④常量（const）等。

（3）在1989年公布的2.0版，又完善了C++的重要概念，形成了更加完善的面向对象的C++程序设计语言：①类的保护成员；②多重继承；③对象的初始化与赋值的递归机制；④抽象类；⑤静态成员函数；⑥const成员函数等。

（4）1993年公布了C++ 3.0版，对C++语言作了进一步的完善，其中最重要的新概念是模板（Template），解决了多重继承产生的二义性问题和相应的构造函数与析构函数的处理等。

（5）1998年，ANSI和ISO先后批准了C++语言标准，使之成了美国国家标准和国际标准。

（6）2000年，C++语言创始人B.Stroustrup推出了《The C++ Programming Language》特别版（Special Edition），其中的C++语言内容根据C++标准进行了更新。

由于C++支持很多高级特性，实现C++的难度加大，所以至今还没有完全支持ANSI/ISO C++标准的编译器出现。相信在不久完全支持C++标准的编译器就会出现。

C++语言是一种不断完善和在实践中改进的语言，在C++ 1.0版发布后的十几年中，C++语言的基本概念仍然在不断地补充和完善。

C++支持面向对象的程序设计方法，特别适合于中型和大型的软件开发项目，从开发时间、费用到软件的重用性、可扩充性、可维护性和可靠性等方面，C++均具有很大的优越性。

C++语言完全兼容C语言，支持多种程序的设计风格，它集中反映了20世纪80年代软件和程序设计技术的全部特征：数据抽象和抽象数据类型、数据的隐藏与封装、支持重用的类机制，类型、变量、函数均可参数化；事前控制程序失败的异常处理机制；支持并发程序设计、多用户系统；支持重载与类型的多态性，以利于程序的扩充；支持重用的继承、派生

机制；支持模块通信的编程风格等。正是这些特征反映了众多的软件新技术，所以，C++语言是应用程序开发者和系统程序员的最佳语言。

3. 由C++语言到Visual C++方法

Visual C++是将可视化编程技术与C++语言的面向对象的程序设计方法相结合所形成的一种全新的程序设计方法。用Visual C++进行程序设计时，可以使用它所提供的功能强大的软件开发工具。在进行程序设计时，甚至不需要手工编写代码，只要根据自己的设计思想，用鼠标或键盘进行操作即可。使得用传统方法很难做到或做起来非常烦琐的工作，用Visual C++都能轻而易举的做到。Visual C++彻底改变了传统的程序设计方法和思维方式，是程序设计人员得心应手的设计工具。

1.2 面向对象的程序设计方法概述

与传统的结构化程序设计语言相比，C++语言是一种面向对象的程序设计语言。

1.2.1 传统的结构化程序设计

在过去的三四十年中，结构化的程序设计（Structured Programming，SP）一直是程序设计开发的主导思想。具体的作法是：为解决某个实际问题确定一个算法，然后为该算法构造合适的数据结构，通过算法的操作过程体现算法的思想，即程序是在数据的某种特定的表示方式和结构的基础上对算法的具体实现，注重的是程序功能的描述。

在面向对象的程序设计技术出现前，程序员一般采用面向过程的程序设计方法。面向过程的程序设计方法用函数（或过程）来描述对数据结构进行的操作，这种方法是将函数与其所操作的数据相互分离。函数作为对现实世界的抽象与它所操作的数据是密切相关并相互依赖的。特定的函数往往要对特定的数据结构进行操作，如果数据结构有所改变，则必须改写相应的函数。这种实质上的依赖与形式上的分离，使用面向过程的程序设计方法编写出来的大型程序，不但难度大，而且难以调试和修改。

结构化程序设计方法力求算法描述准确，对每一个子模块能够容易地进行程序正确性证明，对数据进行编译检查在一定程度上增强了程序的可靠性。这种方法在本质上是面向过程的，不能反映出人们解决问题的思维方法，存在着固有的缺陷，主要表现在以下两个方面。

1. 程序的可重用性差

现在的应用程序越来越复杂，但其中有很多是重复性的工作或差别很小，代码能否重用已成为提高效率的关键所在。在SP模式中即使代码能够重用，也只是简单的复制，稍有不同的地方就必须逐字修改。SP模式不能直接继承引用已编好的应用程序的某些部分，其原因是程序之间的数据传递主要是靠全局变量和指针。

2. 程序的维护和一致性差

由于SP模式是面向过程的，其数据与处理方法是分开的，因此很可能产生数据和方法在结构上的不一致。对程序运行起着重要作用的数据一般要做全局处理，如果为了新的需求而改变某一数据结构的话，所有处理该数据的操作过程都需要重新编写，才能保证与数据的一致性。不但要花费大量的精力，而且还可能产生很多的错误。

1.2.2 面向对象的程序设计

面向对象的程序设计的思维方式与人们日常生活中处理问题的思路相似。在自然界和社

会生活中，一个复杂的事物总是由许多部分组成的。例如，人由五官、身高、体重、学历、政治面目等描述；汽车的基本部件组成由发动机、方向盘、齿轮箱、底盘、车身和车轮等描述；居民住房由客厅、卧室、起居室、厨房和卫生间等组成；学生由小学生、中学生、中专生、大专生、本科生、硕士生和博士生等组成。

当人们生产汽车时，并不是先设计和制造发动机，再设计和制造底盘，然后设计和制造车身和车轮，而是同时设计和制造发动机、底盘、车身和车轮，然后把它们组装在一起。在组装时，各部分之间有一定的联系，以便协调工作。例如，驾驶员踩油门就能调节油路，控制发动机转速，驱动车轮转动。

人们这种对现实事物的认识和分析方法，也是面向对象的程序设计的基本思路。为了克服结构化程序中的弊端，产生了面向对象的程序设计（Object Oriented Programming，OOP）方法，这种方法摆脱了传统过程设计的束缚，使计算机应用程序的开发更符合实际需要。

面向对象的程序设计是以处理的数据为中心，而不是以函数功能为中心编写程序。数据与功能相比具有较强的稳定性，这是面向对象的程序设计与结构化的程序设计的最大区别，前者的中心是所要处理的数据，而后者的中心是函数功能。

面向对象程序设计是围绕事件组织模型的程序设计方法，采用对象来描述问题中的实体。对象是包含现实世界事物特征的抽象实体，反映了系统的信息交互能力。对象是属性及其操作的封装体。在面向对象的程序设计中，“对象”可以认为是“数据＋操作”。

1. 对象的概念

（1）对象：是现实世界中实际存在的任何事务。对象不仅指有形实体，也包括那些抽象的规则、计划或事件等。

客观世界中任何一个事物都可以看成是一个对象（Object），或者说，客观世界是由千千万万个对象组成的。对象可大可小，可以是自然物体（如汽车、房屋、桥梁、熊猫等），也可以是社会生活中的一种逻辑结构（如班级、军队、工厂和政府等），甚至一篇文章、一个图形、一项计划等都可当作对象。

（2）有形实体：一切看得见摸得着的实物。例如，房屋、汽车等是一种易于识别的对象。

（3）无形实体：人类社会中的一种组织或逻辑结构。例如，课程、发展纲要、学校等。

（4）作用：人或组织机构所具有的能力。例如，教师、职工、法院、公司、部门等。

（5）事件：在特定条件下所发生的事情。例如，讲演、开会、打电话、事故等。

对象是面向对象方法中，用于描述客观事物的一个实体，对象的描述是通过对其属性和行为方式进行描述的。对象不仅能表示结构化的数据，而且也能表示抽象的事件、规则以及复杂的工程实体等，这是结构化方法所做不到的。因此对象具有很强的表现功能和描述能力。

在面向对象的设计方法中，对象作为系统中基本的运行实体（实例变量）具有“封装性”。对象封装了描述该对象的特殊属性（数据）和行为方式（方法），从而使对象具有较强的独立性和自治性，其内部状态不受或很少受外界的影响，即对象具有很好的模块化特性。整个应用程序由许多不同类的对象组成，各对象既是独立的实体，又可以通过消息相互作用，对象中的方法决定了要向哪个对象发消息，发什么消息，以及它能收到什么消息，收到消息后如何响应等。

面向对象的设计方法是以数据为中心，这里的“数据”与传统的数据不同，具有“行为”特征。对象的行为是在对象接到消息时发生的，由于对象反映了程序设计领域的模块性，具