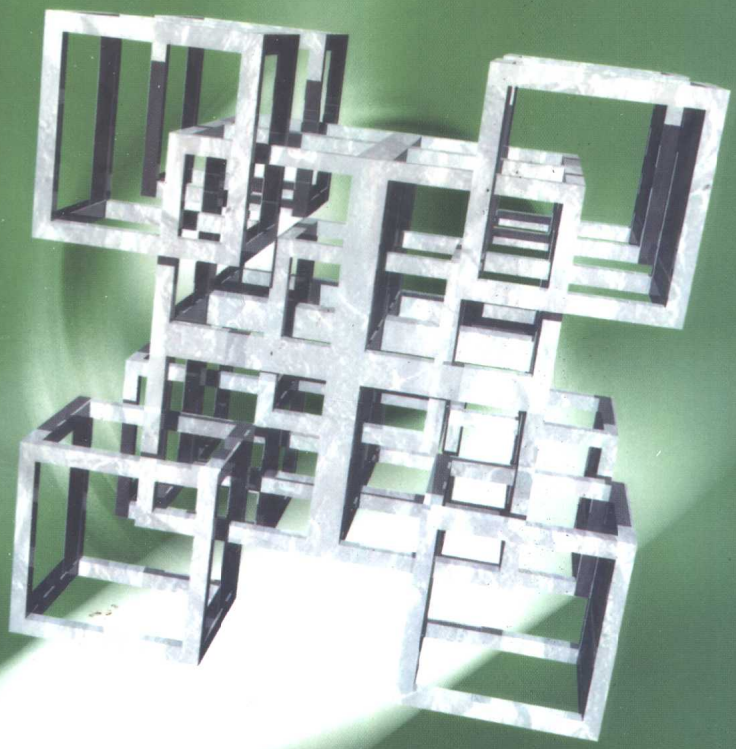
软件编程入门丛书

C++

实用培训教程

杨明军 董亚卓 汪黎 编著



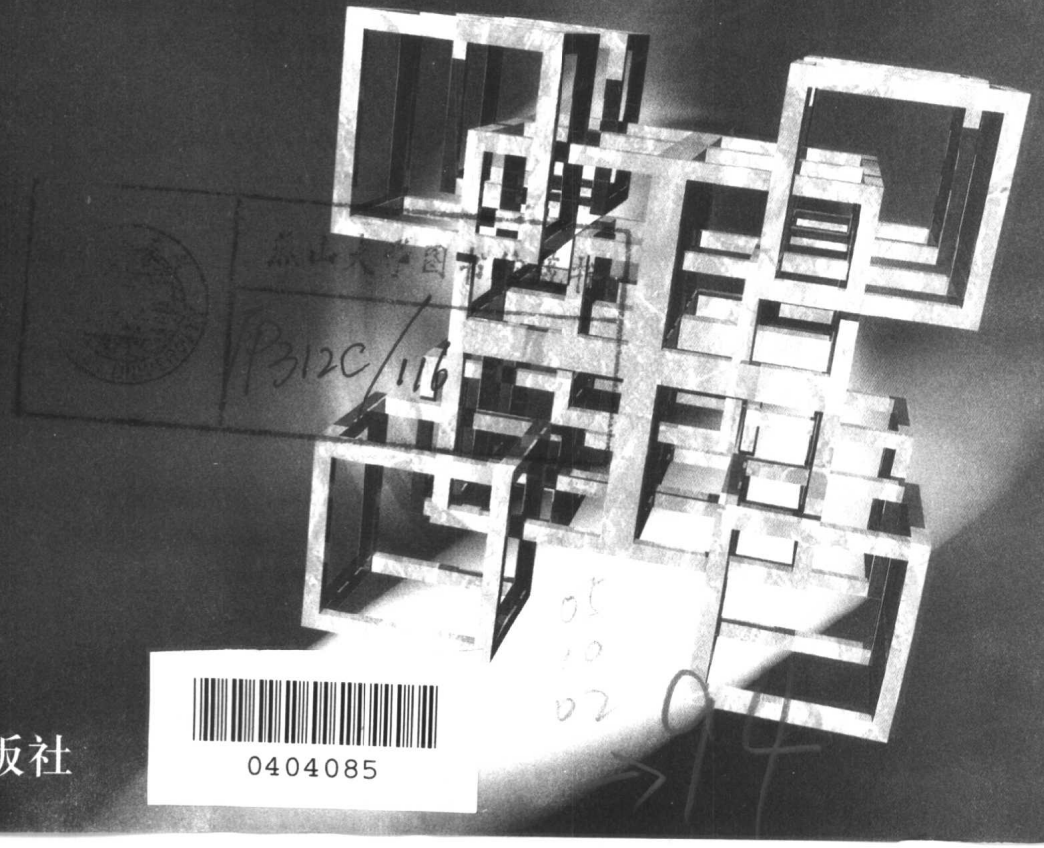
人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

软件编程入门丛书

C++

实用培训教程

杨明军 董亚卓 汪黎 编著



人民邮电出版社

图书在版编目（CIP）数据

C++实用培训教程 / 杨明军，董亚卓，汪黎编著. —北京：人民邮电出版社，2002.12
ISBN 7-115-10763-7

Ⅰ.C… Ⅱ.①杨…②董…③汪… Ⅲ.C语言—程序设计—技术培训—教材 Ⅳ.TP312

中国版本图书馆CIP数据核字（2002）第088794号

内 容 提 要

本书全面系统地介绍了C++语言的实用编程知识。内容包括C++语言基础、类和对象、派生与继承、虚函数与友元、运算符重载、I/O流库和模板与异常等，并在第13章安排了集成开发实例。每章都有实战演练和课后自测，附录中有自测题的参考答案和相关的程序。本书注重基本概念，从实际应用出发，突出重点，叙述清楚，深入浅出，论述详尽。

本书可作为非计算机专业的本、专科高级语言编程教材，也可作为广大电脑爱好者自学C++程序设计方法的指南，以及大专院校和职业学校计算机专业的教学参考。

软件编程入门丛书

C++实用培训教程

◆ 编　　著　杨明军　董亚卓　汪　黎
责任编辑　刘建章　王　艳

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街14号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
读者热线　010-67132692
北京汉魂图文设计有限公司制作
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：22.25
字数：543千字　　2002年12月第1版
印数：1-6 000册　　2002年12月北京第1次印刷

ISBN 7-115-10763-7/TP・3142

定价：30.00元

丛书前言

计算机编程语言是高等学校各专业学生的一门基础课程，也是计算机等级考试的必考科目。但是，随着时间推移，该课程所讲授的知识已越来越落后于时代。大部分的专业或者非专业学生接受的都是 PASCAL、FORTRAN 以及标准 C 语言的基本知识，在得到学分或通过等级考试之后，很少有机会把所学的知识应用于实践中。在走出校门之后，他们马上就面临着必须使用一些以前所不熟悉的编程语言进行工作的任务。

在我国加入 WTO 的大环境影响下，当前国内各类企业对 IT 技术人员的需求日益旺盛，已经远远超过了传统学历教育所能够满足的数量（目前中国的 IT 人才需求为 60 万，缺口则高达 42 万，而每年大学培养的专业人才仅有 5 万）。供需的巨大差距造成软件人才的极度匮乏。软件人才的缺乏将成为我国信息技术发展的又一瓶颈。有关专家介绍，软件人才短缺的主要根源是软件教育体制调整的速度落后于软件产业发展的速度，教材陈旧、教育理论与实践脱节、学生英文水平低等因素进一步加剧软件人才的短缺。

这就需要通过各种途径，比如短期培训和继续教育，为这一行业的发展提供大批能够基本掌握编程知识的 IT 技术人材。考察对这些编程人员的需求可以发现，其中的大部分任务都只是要求处理日常事务，并不需要太多的编程思想、算法和逻辑等专业知识。

随着计算机技术在各个领域的广泛应用，以及编程环境可视化程度的不断提升，计算机编程早已不再是计算机专业人员的特定任务。由于业务工作需要或者提高自身素质的要求，或主动或被动地，越来越多的计算机爱好者通过专项培训或上机自学，将会涉足到计算机编程这个原本让人觉得高不可攀、神妙莫测、敬而远之的领域中来。

本丛书就是为此目的而编写的，它以计算机编程为核心，涵盖了从基础知识到专业应用的一系列重要内容。在内容组织上狠下功夫，全书虚拟课堂教学模式，每章基本上都划分为“教授主讲”、“实战演练”、“学以致用”、“课后自测”4 小节。按照理论→实践→提高→巩固的主线，采用 Step by Step 的讲解方式，结合实例的设计思路，进行创意与扩展，使读者学用轻松。本书旨在达到学以致用的目的，内容叙述由浅入深、循序渐进，适用于编程的初、中级用户；同时，一些大程序的开发过程及编程技巧，对有经验的程序员来说也有很好的参考与借鉴价值。

本丛书具有如下特点：

- 深入浅出通俗易懂

  面向的读者不只是计算机专业人员，更为重要的是面向计算机编程爱好者和编程培训班学员，因此在内容安排和文字叙述上尽量深入浅出、通俗易懂，力求讲清楚问题的来龙去脉，能够让读者清晰地明白编程的“过程”。

- 将编程思想与开发工具的运用紧密结合

  在学习编程的过程中，不仅要在学习编程思想上有所突破，还应学会如何更好地运用编程

的开发工具，只有两者的结合才是真正的理论联系实际、事半功倍的学习方法。本丛书精选了目前流行的软件开发工具，对编程者具有实际的应用价值。

◆ 理解编程的实质

作为一个编程人员，必须强化编程的概念从而理解编程的实质，才能做到举一反三融会贯通，才能编制自己的应用程序。所以本丛书并不是着力去逐条讲解语句和库函数的使用，而是针对学习中可能遇到的问题，讲解分析思路和编程技巧，力求提高编程能力。

◆ 重视习题实战训练

如果一本编程的书不能启发式地让读者试图将所学知识运用于日常工作，那么就没有很好地完成教学任务。本丛书的每章后面都提供了大量的习题，而且在书的最后给出了这些习题的答案提示，从而达到快速掌握编程方法和技巧的目的。

为了真正地实施精品策略，认真编写好这套教程，我们在各级高校、社会办学机构、编程培训班和数家大型公司进行了广泛、系统而详细的调查，邀请在教学、科研和工程第一线中富有培训和实践经验的大批学者、专家和教授参与编写，多次组织由专家和高校一线教师参加的研讨会，对现有图书市场上的类似教程进行综合分析和优缺点对比以博采众长，以求达到理论知识性和实用工程性的完美结合。

本套丛书的全部源代码和一些相关的详细资料，都可以从人民邮电出版社计算机图书第一出版中心网站（www.ucbook.com）上打包下载。由于我们的水平和经验有限，这批教程在编写、审查和出版工作中肯定还存在不少的缺点和不足，希望使用本套教程的读者提出批评和建议，以便改进我们的工作，让教程的质量不断地提高。

编者

2002 年 10 月

前　言

随着社会信息化程度的不断提高，对各类计算机人才的需求也不断增长。目前国家不仅需要大量的 IT 业从业人员，在其他各行业也同样需要具有较高计算机应用水平的复合型人才。为了适应这种迫切的需要，促进我国计算机知识的普及，提高全社会的计算机应用水平，我们编写了这套《软件编程入门丛书》，本书即为丛书之一。

在各种领域的应用软件设计中，C++已经成为不可或缺的工具；在各行各业的招聘面谈中，能否熟练掌握 C++语言已经成为择优录取的重要条件。过去，人们以为只有计算机专家才会 C++，只有计算机专业才需要学习 C++；现在，这种观点已经与现实世界大相径庭。

许多人误以为面向对象的概念十分抽象，什么封装、继承，太难懂了。其实，这完全是一个误导。面向对象这个革命性的计算概念，是今天大多数高级语言的基石，C++的基石是面向对象，Java 的基石也是面向对象。C++是一种面向对象程序设计语言，但不是纯面向对象语言，而是混合型面向对象语言，它在过程语言中增加了面向对象的结构。这种特性使得 C++保持与 C 语言的兼容，使许多 C 语言代码不经修改就可以为 C++使用。

我们在构思本书结构的时候，不希望将 C 语言面向过程的内容和 C++面向对象的内容截然分开来讲，这样大家所学到的东西往往仍是用 C 语言写的所谓 C++程序。我们希望读者在学习基础知识的过程中也能够潜移默化地接受 C++的思想，比如在第 1 章“准备开始”中就已经使用 C++的 I/O 语句 cin 和 cout，在第 2 章的“移位操作符”一节中埋下“操作符重载”的伏笔，在第 4 章“函数”中就及时引入“函数的重载”概念，这些都是和 C 语言截然不同的语法。

之所以在讲解数组、字符串、指针等 C++特性之前就引入对象和类的概念，是为了让读者早日接触到面向对象的思想，以便于在后文中灵活使用。在“数组和字符串”一章中已经使用了对象数组和自定义的字符串类，而在“指针”一章中更有对象指针、this 指针等需要对象和类的知识。这样的安排，是为了让读者能够将对象和类看成是和简单数据类型（比如整数、小数等）一样的用法，有利于消除读者对于“对象”高不可攀的神秘印象。其实，简单的对象往往比数组还要容易使用。这是 C++的核心所在，早点掌握可能会更合适一些。

这是一本面向非计算机专业的计算机入门教程，同时也可以作为计算机专业的教学参考书。本书读者并不需要一定有 C 语言的基础，但是如果稍微有一些基本的编程语言常识，可能会有事半功倍的效果。在讲解和举例的过程中，我们尽量只使用前文中讲到过的概念和语法，以求能够循序渐进；可是由于 C++的博大精深和高度灵活，很抱歉可能书中仍然存在先使用再定义的情况。读者在遇到一时看不懂的东西时，可以暂时将它搁置一边继续向下读，稍后再回头就会有豁然开朗的感觉。

在写作的过程中，我们力图让读者从对 C/C++一无所知开始，迅速迈入 C++编程的殿堂。

本书的写作风格特别适合于自学，当作听课的教材也未尝不可。它细腻而不失于简练，翔实而不落于烦琐，连贯而不流于形式。每章后面都有精心设计的练习，从相当简单的概念分析，到大多数人都要花数个小时的编程项目，让读者能验证自己知识掌握的程度。除了阅读程序题之外，希望读者能从中选择一部分作为上机练习，独立完成这些习题，必将极大地提高自己的程序设计能力。

本书是笔者在总结多年从事C++语言研究、应用和教学与实践经验的基础上，参考国内外有关资料编写而成的。精心挑选了一些现实中应用的典型问题作为本书的例题，同时也特别注意了程序结构和程序设计方法。每个例题都在Turbo C++ 3.0、Borland C++ 3.1或Visual C++ 6.0环境下调试通过。虽然我们力图精益求精以求完美，但是由于时间仓促与水平有限，书中难免有疏漏之处，还望广大读者不吝赐教，或发E-mail到huben@nudt.edu.cn处与我们联系。

编者

2002年10月

目　录

第 1 章　C++编程准备 1
1.1　教授主讲 1
1.2　实战演练 5
1.2.1　第一次尝试 5
1.2.2　C++程序的组成部分 5
1.2.3　C++单词 8
1.2.4　运行C++程序 9
1.3　学以致用 10
1.3.1　经验借鉴 10
1.3.2　自我理解的开端 11
1.4　课后自测 12

第 2 章　数据类型和表达式 13
2.1　教授主讲 13
2.1.1　C++基本的数据类型 13
2.1.2　常量和变量 14
2.1.3　运算符 18
2.1.4　表达式 24
2.1.5　类型转换 28
2.2　实战演练 31
2.3　学以致用 32
2.4　课后自测 32

第 3 章　循环和选择 33
3.1　教授主讲 33
3.1.1　关系运算符与逻辑运算符 33
3.1.2　循环 34
3.1.3　选择 37
3.1.4　其他控制语句 42
3.1.5　预处理 44

3.1.6 条件编译 47
3.2 实战演练 50
3.3 学以致用 52
3.4 课后自测 52

第 4 章 函数 53
4.1 教授主讲 53
4.1.1 函数声明 54
4.1.2 函数调用 56
4.1.3 函数的参数 60
4.1.4 函数重载 61
4.1.5 函数的嵌套调用和递归调用 64
4.1.6 内联函数 67
4.1.7 函数的作用域 69
4.1.8 C++的库函数 74
4.2 实战演练 76
4.3 学以致用 79
4.4 课后自测 79

第 5 章 对象和类 81
5.1 教授主讲 81
5.1.1 从现实中抽象 81
5.1.2 类的三大特性 82
5.1.3 定义类 82
5.1.4 类的使用 88
5.1.5 类的构造函数 89
5.1.6 拷贝构造函数与赋值函数 94
5.1.7 类的析构函数 97
5.1.8 对象的作用 97
5.1.9 类与结构 99
5.2 实战演练 100
5.3 学以致用 102
5.4 课后自测 103

第 6 章 数组和字符串 105
6.1 教授主讲 105

6.1.1 数组使用基础 105
6.1.2 作为类成员数据的数组 115
6.1.3 对象数组 116
6.1.4 C++字符串 118
6.2 实战演练 126
6.3 课后自测 128

第 7 章 指针 129
7.1 教授主讲 130
7.1.1 指针和地址 130
7.1.2 指针赋值 131
7.1.3 字符指针 133
7.1.4 指针和数组 134
7.1.5 类成员指针 141
7.1.6 对象指针 142
7.1.7 指针和函数 143
7.1.8 this指针 146
7.1.9 带指针的main()函数 147
7.1.10 内存管理：new和delete 148
7.2 实战演练 152
7.3 学以致用 154
7.4 课后自测 154

第 8 章 派生和继承 155
8.1 教授主讲 155
8.1.1 派生类和基类 156
8.1.2 类的层次性 157
8.1.3 单继承 158
8.1.4 多继承 169
8.2 实战演练 176
8.3 学以致用 180
8.4 课后自测 181

第 9 章 虚函数和友元 185
9.1 教授主讲 185
9.1.1 虚函数 185

9.1.2 抽象类.....198
9.1.3 友元.....200
9.1.4 引用.....208
9.2 实战演练.....214
9.3 课后自测.....216

第 10 章 运算符重载.....219
10.1 教授主讲.....219
10.1.1 重载运算符.....219
10.1.2 运算符作为成员函数.....224
10.1.3 一些特殊运算符的重载.....228
10.1.4 数据转换.....234
10.2 实战演练.....240
10.3 学以致用.....244
10.4 课后自测.....244

第 11 章 I/O 流和输出.....245
11.1 教授主讲.....245
11.1.1 标准I/O.....245
11.1.2 文件I/O.....260
11.1.3 串I/O.....270
11.2 实战演练.....272
11.3 学以致用.....274
11.4 课后自测.....274

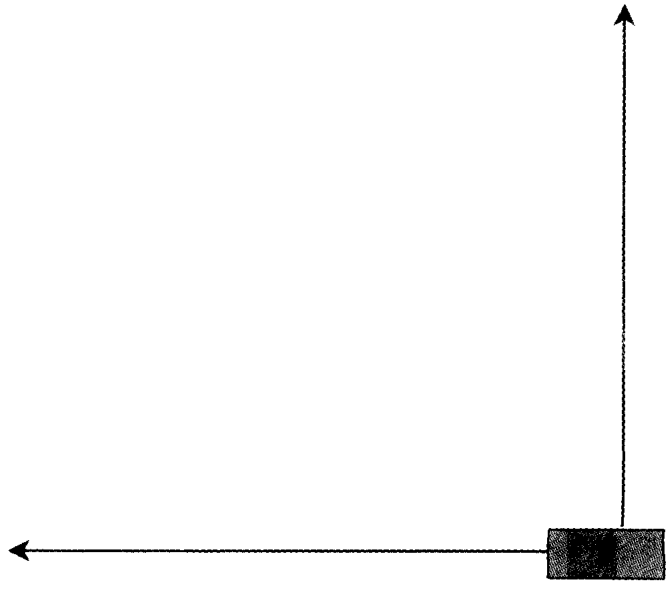
第 12 章 模板和异常.....275
12.1 教授主讲.....275
12.1.1 模板基础.....275
12.1.2 模板分类.....277
12.1.3 类模板.....278
12.1.4 异常的概念.....281
12.1.5 异常处理的实现.....282
12.1.6 标准异常.....286
12.1.7 异常程序设计规则.....287
12.1.8 小结.....288
12.2 实战演练.....288

12.3 课后自测 294

第 13 章 集成开发实例 295
13.1 以一个简单的小游戏开始 295
13.2 和电脑对战五子棋 299
13.2.1 主函数流程分析 299
13.2.2 每个类的分析 304
13.3 推荐书目 315

附录 A 部分习题解答 317

附录 B 综合上机测试 329



第1章 C++编程准备

课前导读

C++程序设计语言是当今版本较新、功能强大的语言之一。它基于C语言，又大大发展了C语言。最突出的是，C++提出了类（class）的概念。类是数据和函数的集合，数据用来描述类所定义的对象的状态，函数用来描述此类定义对象的行为。例如，学生代表一类人，即学生是个类，每个具体的学生都是这个类中的对象。这个类中的数据可以是学生的姓名、性别、年龄、学校等，描述此类对象的行为可以是入学、入党、毕业等。在C语言中，结构只是数据的集合，C++将C语言中的“结构”概念扩充成上述“类”的概念，即既可以有数据，也可以有函数的“类”。

本书会让读者从最基本内容的开始，知道应该学习什么，应该怎样学习，应该如何训练，如何培养好的编程习惯，如何以本书为跳板快速跨越C++那高高的门槛。

要点提示

本章将主要介绍下面一些内容：

- 从C到C++的演变
- 程序的基本结构框架
- C++成功的奥秘
- 面向对象程序设计的基本特征：对象、类和继承
- C++的单词

1.1 教授主讲

下面将一步一步从头讲起，先让你很快地对你的选择有个概括的了解，培养一种亲密感。

1. 从C到C++的演变

要学好C++，还真的不能忽略C语言的作用，至少你应该了解这位C++的“父亲”。在1970年，Ken Thompson根据BCPL（Basic Combined Programming Language）语言设计出较先进的并取名为B的语言，随后C语言问世了。C语言是1972年由美国的Dennis Ritchie设计发明的，并首次在UNIX操作系统上使用。由于没有统一的标准，出现了许多C语言版本。这些不同版本的C语言之间出现了一些不一致的地方，使程序的可移植性受到极大阻碍，制约了C语言的推广。随着微型计算机的日益普及，这种缺点也日益突出。

为了改变这种情况，美国国家标准研究所ANSI（American National Standard Institute）

为C语言制定了一套ANSI标准，成为现行的C语言标准。之后在1987年有了Turbo C，现仍在学校中广泛使用。Turbo C是美国当时著名的软件公司Borland的产品，它使用了焕然一新的集成开发环境，即使用了下拉式菜单，将文本编辑、程序编译、连接以及程序运行一体化，大大方便了程序的开发。随后Turbo C 2.0版本问世，增加了图形库和文本窗口函数库等加强组件，并且初步包含了C++中极为重要的“面向对象”的基本思想和设计方法。1991年为了适用于Microsoft公司的Windows 3.0版本，Borland公司又将本书的主角Turbo C++ 3.0 IDE推向市场，而Turbo C的新一代产品Borland C++也问世了。

追根溯源，C++的产生还更早，由一位来自AT&T贝尔实验室的Bjarne Stroustrup设计和实现的。C++最初的版本被称作“带类的C（C with classes）”[Stroustrup，1980]，在1980年被第一次投入使用；当时它只支持系统程序设计和数据抽象技术。在1983年向C++加入支持面向对象的程序设计思想之后，类才真正进入了C++领域。随着独立开发的C++产品的出现和广泛应用，C++标准化工作在1990年开始启动。标准化工作由ANSI以及后来加入的ISO（International Standards Organization）负责。经过漫长的9年，直到1998年才正式发布了C++语言的国际标准[C++，1998]。在这期间，C++的发展如日中天，为计算机产业的发展立下了汗马功劳。

C++源于C语言，严格地说，它是C语言的一个超集：几乎所有C中正确的语句在C++中也同样是正确的，反之则不成立。添加进C的最重要的成分是能够帮助创建C++所关注的类、对象和进行面向对象的程序设计。此外，C++还有很多自己的特性，如改进了的输入/输出（I/O）和采用了一种写注释的新方法。C与C++的关系如图1-1所示。

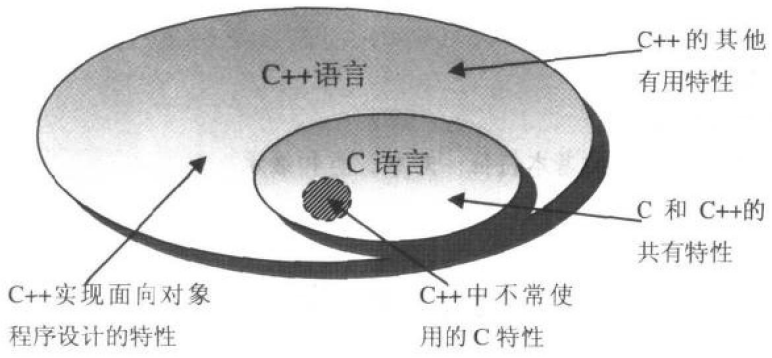
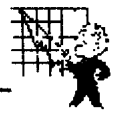


图1-1 C与C++的关系

2. 程序

计算机再神秘也只是一台机器，它不会思考，也没有感情，只能执行人们编写的程序。程序是告诉计算机做什么的一组指令，用以表达人们的思想，是以某种语言为工具编制出来的动作序列。它是有目的性的，只能实现预先想好的功能。因此，它是一种软件。

假设计算机是一台电视，程序就好比遥控器。你想看什么电视节目，还需要自己去选台，否则它什么也干不了。只是，计算机程序要强大很多，对你所做的指示，可以不折不扣地执行。当然，你不能让它去帮你吃午饭，让它模拟你吃午饭的动作还是可能的。



计算机程序是按照计算机程序设计语言所要求的规范书写出来的一系列动作。对于计算机来说，一组机器指令就是一个程序；对于使用计算机的人来说，用某种高级语言编写的语句序列就是程序。所以，源文件、源程序、源代码都是程序。

随着程序变得越来越大而且更为复杂，结构化程序设计的方法显得捉襟见肘。你可能听说过，或者经历过程序开发中的恐怖故事。项目非常复杂，进度延期，增加程序员，复杂度增加，成本直线上升，进度再延期，然后灾难降临（有兴趣的读者可以参考《人月神话》[The Mythical Man-Month]，书中对这个情形的描述非常生动。）。

分析这些失败，其原因往往显示出错在面向过程的本身。不论结构化程序设计方法实现得多么好，大型程序都会变得过于复杂。面向过程语言会出现这些问题的原因是什么？有两个相互关联的问题：首先，函数能够不受限制地访问全局性数据；其次，函数和数据之间缺乏联系，面向过程为现实世界提供的模型可以称得上粗劣。

3. C++成功的奥秘

前面已经说到，C++在软件界的成功是无可非议的。C++之所以能够如此成功，部分原因是它的目标不只是为了将 C 语言转变成面向对象的程序设计语言（简称为 OPP），而且还为了解决当今程序员，特别是那些在 C 语言中已经投入大量时间和心血的程序员们所面临的许多问题——C 语言所不能完成或难以完成的任务。C 语言所倡导的结构化语言，使得程序员能非常方便地编写出有一定难度的程序。然而，一旦项目达到某一程度，使用结构化的编程方法就变得难以控制，其复杂度往往使得富有经验的编程高手也感到头痛。

比如，当一个大型程序的数据项被修改时，很难确定有哪些函数访问了这个数据；甚至即使将它们找出来了，但接下来的修改却使得函数不能与其他的全局数据项一起正常工作。每一个事物都与其他的事物相互联系，以致任何地方的一个修改都能影响到其他的地方，而且常常伴随着无法预料的后果。

面向对象编程吸取了结构化编程的最佳思想，并把它们与几个强有力的概念结合在一起，以一种新的方法完成编程任务。一般来说，当用面向对象的方法编程时，可以把一个问题分解成几个彼此相关的部分，注意到每个部分相关的代码和数据，再把这些部分组织成一个层次结构，最后组织成一个自包含单元。

C++的目的是提高效率。“工欲善其事，必先利其器”，语言这种抽象的工具是为了尽可能地帮助使用者，尽可能不用武断的规则或特殊的性能妨碍使用者。C++之所以成功，是因为它立足于实际：尽可能地为程序员提供最大便利。

4. 面向对象程序设计的基本特征

面向对象的系统包含了 3 个要素：对象、类和继承。

（1）对象

面向对象语言的最重要特征是对象。简而言之，对象是一个逻辑实体，它含有数据及处理该数据的代码。在一个对象中，有些代码和数据可能是由该对象专用的，不得由该对象外的任意成分存取，好比你存在银行的钱，你是惟一的支配者。这样的话，对象提供一个有意义的保护级，以防止程序中不相关的部分被偶尔修改，或不正当地使用该对象的私用部分。

代码和数据的这一连接常被称为封装，比如你的钱封装在你的银行帐号里。

事实上，对象是用户定义类型的变量。就好比你的钱可以存为人民币，也可以兑换成美元再存入银行。对象的表示可以变，但所代表的本质不变，你的钱可以兑换成不同的货币，但它所代表的真实价值却不会增加。初看它有点怪（连接代码和数据的对象看作变量），然而在面向对象编程中，这是极为精确的表达形式。当定义对象时，便隐式地建立了一个数据类型。在此说起来显得很抽象，你需要在后面不断的学习中积累消化。

（2）类

类是一种较为复杂的数据类型，它是将不同类型的数据和与这些数据相关的操作封装在一起的集合体。类具有更高的抽象性，类中的数据具有隐藏性。类还具有封装性。

类的结构（即类的组成）用来确定一类对象的行为，而这些行为是通过类的内部数据结构和相关的操作来确定的。这些行为是通过一种操作接口来描述的（也即平时我们所看到的类的成员函数），使用者只关心接口的功能（也就是我们只关心类的各个成员函数的功能），对它是如何实现的并不感兴趣。操作接口又被称为这类对象向其他对象所提供的服务。

就类似于我们在使用电视机的时候，对电视机上的每一个控制键的功能可能了如指掌，闭着眼睛也能找到调音量、选台等按钮，可是你能说清楚这个按钮为什么能实现这个功能吗？应该有很多人不能回答吧。可能你会说："即使不知道，也能熟练地控制电视，随心所欲地找到自己要看的节目呀。"没错，对于一个只用电视来休闲娱乐的使用者来说，熟练使用就已经足够了，没有必要了解得更多。C++中的类正是借鉴了这种思想，只需关心功能接口，而无需了解具体实现。这样在编写程序时，就没有必要事事从头做起。对某些已经完成的模板便可实施"拿来主义"，因为它们已经实现了一些特定的功能；至于是怎样实现的呢，我们不必关心，只要会用就行。这才是一个聪明的学习者。

（3）继承

对于现实生活中的绝大多数知识，都可以运用层次分类这一概念。例如：龙眼葡萄属于葡萄类，葡萄又属于水果类，而水果又属于更大的食品类。生物系统中形形色色的生物，也可以划分为界、门、纲、目、科、属、种。如果不使用分类，就不得不用显示式来定义每个对象的所有特性。通过使用分类概念后，描述一个对象便仅需定义那些使其在类中惟一的属性。这便是一种继承机制，正如万物都有本源，很多事物都是从一种特定模式中继承出来的，但同时事物又各有各的特性而千差万别。所以只要把事物的共性抽象出来，再针对具体的事物定义它的个性，这样对自然界的定义就变得简单了，同时也符合了自然规律。继承使对象有可能成为更一般情况的一个特定事例。

准备工作就此完毕。通过以上阅读，你应该对C++的特点有一个初步的了解；同时，你应该认识到编写程序并不神秘，完全是按照人的正常思维逐步进化完善的。如果对某些概念还不是很理解，不必着急。现在你还只读到了第4页而已，相信在读到第40页的时候，就会对它们有深刻的体会。下面就让我们进入真正的C++程序。

1.2　实战演练

1.2.1　第一次尝试

准备好，开始看你的第一个 C++程序：

例 1-1　C++的一个示范程序

```
//This is a simple C++ program
#include<iostream.h>
int main()
{
    double a,b;
    cout<<"Enter two float numbers:";
    cin>>a>>b;
    double c=a+b;                    /*  相加求和    */
    cout<<"a+b="<<c<<endl;
    return (1);
}
```

感觉怎么样？这是一个很简单的 C++程序，在这个程序中，只有一个函数 main（），函数体内有 5 条语句。下面我们就把这个程序一层层地剥开，了解一下 C++程序的组成部分。

1.2.2　C++程序的组成部分

在 C++中，每一个行为被描述成一个表达式，无论多小的一句话，只要以一个分号结束的表达式都被称为一个语句。在 C++中，最小的独立单元是语句。如

```
;     //空白语句，延时作用
```

具有语句表示所需的终结符“;”作结尾，也被称作一条语句。

1. 预处理指令

C++中提供了 3 种预处理指令：宏定义命令、文件包含命令和条件编译命令。例 1-1 中出现的预处理命令是文件包含命令：

```
#include<iostream.h>
```

其中，include 是关键字，尖括号内是被包含的标准函数。文件被存放在标准的头文件中，例如程序中用到的 iostream.h。在头文件中保存着所有与标准函数有关的信息。为了使用头文件中定义的变量和函数，我们必须将相应的头文件包含进主程序，使头文件成为自己程序中一部分的指令是 include 预处理指令。所有的预处理指令都由“#”引导，且“#”号必须置于该指令行的开始。预处理指令在程序编译前执行，其执行结果以中间文件的形式传递给编译