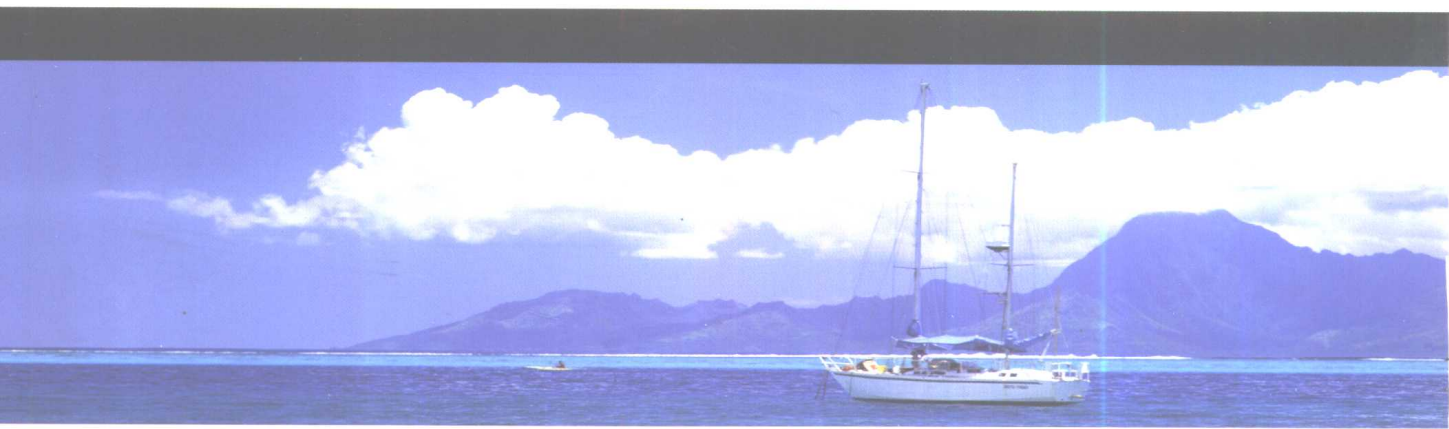


21世纪高职
计算机系列教程

C/C++ 程序设计



Decoder 编著

引进台湾原版成熟教材■

注重内容的实用性，培养学生的专业能力■

适用于高职电子信息类各专业选用■

经过全国高等院校计算机基础教育研究会高职高专专业委员会■

教授及一些著名专家学者的评估与审定

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

高职系列教材

号 8800-5005-10 :号51登同合社中著局社出

种 程 序 册

C/C++程序设计

Decoder 编著

中国铁道出版社

2002年·北京

(京)新登字 063 号

北京市版权局著作权合同登记号: 01-2002-0088 号

版 权 声 明

本书中文繁体字版由台湾第三波电脑图书资料股份有限公司出版, 2002。本书中文简体字版经台湾第三波电脑图书资料股份有限公司授权由中国铁道出版社出版, 2002。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

本书封底贴有台湾第三波电脑图书资料股份有限公司防伪标签, 无标签者不得销售。

图书在版编目 (CIP) 数据

C/C++程序设计/台湾慷齐咨询有限公司编著. —北京: 中国铁道出版社, 2001. 12

ISBN 7-113-04500-6

I. C… II. 台… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 096765 号

MS2P8/03

书 名: C/C++程序设计

作 者: Decoder

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 苏 茜

特邀编辑: 袁秀珍

封面设计: 孙天昭

印 刷: 北京兴顺印刷厂

开 本: 787×1092 1/16 印张: 24.5 字数: 587 千

版 本: 2002 年 2 月第 1 版 2002 年 2 月第 1 次印刷

印 数: 1~ 5000 册

书 号: ISBN 7-113-04500-6/TP·661

定 价: 29.00 元

版权所有 盗印必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

21 世纪高职高专系列教材 审定委员会

主任：高 林

副主任：丁桂芝 李 畅

委员：安淑芝 鲍有文 陈文博 樊月华 陈维兴
顾星海 李秀芬

项目策划：严晓舟 宋旭明

编 辑：郭晓溪 郭毅鹏 裴剑波

前 言

随着社会经济的高速发展，我国的高等教育已步入从精英教育走向大众化教育的发展阶段。国际高等教育的历史说明高等职业教育必将成为这一发展过程的主力军。当前从专科起步的高等职业教育，即高职高专教育还处于探索阶段，教材建设已成为教育改革的重要方面。

高职高专教育是高等教育的一个新类型，它与传统的普通高等教育既有紧密地联系，又有本质的区别，高职高专教育强调面向社会、生产、管理、服务第一线，培养技术应用性人才，学生毕业后，即可发挥其所学专长。所以，根据我国教育部规定，高职高专教育不得使用普通高等教育教材，而要根据高等职业教育的特点，建立自己的教材体系。

我国台湾地区的高等职业教育发展已相当成熟，在过去的二十几年中，为台湾地区经济发展培养了大批应用性人才，在职业化教育方面积累了丰富的经验；通过多年的实践在处理理论与实践的关系以及培养实际操作的技术应用性人才方面都处于领先地位，也符合国际职业教育发展的主流。由于有着相同的历史文化背景，现阶段台湾的高等职业教育教材对我们更具有借鉴作用。

因此根据教育部关于引进外文原版教材、推动我国教材改革的精神，通过多方面调查，在充分听取专家意见的基础上，中国铁道出版社以“引进，吸收，创新”为指导思想，力争走出一条新的高职高专教材发展之路。作为这一决策的第一步，我们经过精心的组织策划，推出了“21世纪台湾版高职高专系列教材”。

我们本次引进的教材都是现阶段我国台湾地区的科技大学和技职学院正在使用的教材，这些学校是当前台湾发展高等职业教育的主体学校，我们认为这些教材在编写过程中均采用了理论与实际相结合的方法，体现了能力本位的思想，注重在学习理论知识的基础上重点培养学生的实践能力，通过一系列实际训练，提高学生各种必备的技术应用能力，使他们一出校门便可发挥其所学专长，成为该领域的专门人才。

本套教材一开始便受到了全国高等学校计算机基础教育研究会高职高专委员会的高度重视，委员会主任高林教授充分肯定了铁道出版社引进台湾版高职教材的思路，并在教材引进和编审过程中给予了全面指导，组织多位专家教授对这套教材进行了审定。专家一致认为本套教材内容新颖，易教易学，层次配套，符合高等职业教育的特点，有利于推动我国高等职业教育的发展，建议在全国范围内推广使用。

中国铁道出版社
2002年2月

出版说明

本书从 C++语言的基础开始，为您打下专业的 C++面向对象程序设计根基。然后以完整的 C++指令程序练习搭配范例，教您设计面向对象程序，以及撰写可重复使用的类。书中的程序除了赋予流程图之外，更注入程序设计的观念，让学习者通过轻松的学习流程，学会 C/C++的语法，借助完整的程序解析，学会范例应用的程序。更可了解 C++面向对象程序设计的方法、经验，彻底学会 C++程序设计。

书中范例程序可在 <http://www.tqbooks.com> 天勤网站下载。

本书由第三波资讯股份有限公司提供版权，经中国铁道出版社计算机图书项目中心审选，王秀平、李自运、梁秀玲、关超、史广顺、陈辑超、马超、杨小平、段小明、杨军、陈贤淑及孟丽花等同志参与了本书的整稿及编排工作。

2002 年 2 月

目 录

第 0 章 认识 C++	1
0-1 关于面向对象	2
0-2 为什么要使用面向对象	2
0-2-1 浅谈传统的数据处理	2
0-2-2 面向对象的数据处理方式	3
0-3 面向对象的优点	3
0-3-1 对象	3
0-3-2 类	4
0-3-3 继承	4
0-3-4 重用	5
0-4 C++与 C	5
第 1 章 第一个 C++程序	7
1-1 Visual C++的程序编辑环境	8
1-1-1 编辑器窗口	8
1-1-2 编译器	9
1-2 基本的程序结构	10
1-2-1 函数 (Function)	10
1-2-2 输出与输入	11
1-2-3 预处理程序	12
1-2-4 程序的注释	12
1-3 编写第一个程序	13
第 2 章 变量与数据类型	17
2-1 浅谈数据类型	18
2-2 整型 (Integer)	18
2-2-1 短整型	19
2-2-2 长整型	19
2-2-3 整型类型的修饰符	20
2-3 字符 (Character)	21
2-4 浮点数 (Floating Point)	22
2-5 布尔 (Boolean)	24



2-6	unsigned 数据类型	25
2-7	变量与常量	26
2-7-1	变量	26
2-7-2	常量	28
2-7-3	变量与常量的生命周期	29
2-8	转换数据类型	32
2-8-1	自动转换	33
2-8-2	强制转换	35
第3章	运算符	37
3-1	赋值运算符	38
3-1-1	概念解析	38
3-1-2	范例程序	38
3-2	算术运算符	40
3-2-1	运算符介绍	40
3-2-2	更简洁的语句方式	43
3-2-3	前缀与后缀记法	46
3-3	关系运算符	48
3-3-1	概念解析	48
3-3-2	范例程序	49
3-4	逻辑运算符	51
3-4-1	概念解析	51
3-4-2	范例程序	52
3-5	位运算符	53
3-5-1	And 位运算符	54
3-5-2	Or 位运算符	59
3-5-3	按位异或运算符	60
3-5-4	Not 位运算符	64
3-5-5	位移运算符	64
3-6	其他运算符	66
3-6-1	条件运算符	67
3-6-2	逗号运算符	67
3-6-3	求字节数运算符	67
3-7	运算符的优先级	68
第4章	决策与循环	69
4-1	循环	70
4-1-1	概念解释	70
4-1-2	范例程序	71
4-2	while 循环	72

4-2-1	概念解释	72
4-2-2	范例程序	73
4-3	do...while 循环	74
4-3-1	概念解析	74
4-3-2	范例程序	75
4-4	嵌套循环	76
4-4-1	概念解析	77
4-4-2	范例程序	77
4-5	if 语句	79
4-5-1	单纯的 if 语句	79
4-5-2	if...else 语句	82
4-5-3	else If 语句	84
4-6	switch 语句	89
4-6-1	概念解析	89
4-6-2	范例程序	90
4-7	转向语句	92
4-7-1	break 语句	93
4-7-2	continue 语句	94
4-7-3	goto 语句	97
第 5 章	函数	101
5-1	简单的函数	102
5-1-1	函数的声明	102
5-1-2	函数的定义	103
5-1-3	调用函数	106
5-2	参数与返回值	111
5-2-1	声明函数	111
5-2-2	定义函数	112
5-2-3	传入参数给函数	113
5-2-4	函数返回值	118
5-3	传址参数	122
5-4	默认参数值的函数	124
5-5	内联函数 (Inline Function)	126
5-6	变量的种类	127
5-6-1	自动变量与静态变量	128
5-6-2	局部变量与全局变量	130
第 6 章	数组	133
6-1	使用数组的好处	134
6-2	一维数组	135



6-2-1	数组的声明	136
6-2-2	数组的初始化	137
6-2-3	数组的元素访问	138
6-3	多维数组	141
6-3-1	二维数组的声明	142
6-3-1	二维数组元素访问	143
6-4	将数组当作函数参数	145
6-5	字符串	150
6-5-1	基本概念	150
6-5-2	字符串运行	151
6-5-3	逆转字符串	154
6-5-4	string 类	156
6-6	数组的应用	160
6-6-1	数组与矩阵	160
6-6-2	排序	162
6-6-3	栈	165
第 7 章	结构与枚举	171
7-1	结构的定义	172
7-1-1	概念解析	172
7-2	结构成员的访问	176
7-2-1	范例程序一——员工资料的结构	177
7-2-2	范例程序二——日期时间的结构	178
7-3	嵌套结构	181
7-3-1	概念解析	181
7-4	传递结构变量给函数	183
7-5	函数返回结构变量	183
7-6	结构数组	185
7-7	枚举的定义	188
7-7-1	枚举的语法	189
7-7-2	范例程序——判断及响应	189
第 8 章	指针	195
8-1	指针与内存	196
8-1-1	预备知识：二进制与十六进制	197
8-1-2	内存地址	198
8-1-3	地址运算符"&"	199
8-2	指针的声明	202
8-2-1	指针变量	202
8-2-2	指针常量	207

8-2-3	指向指针的指针	208
8-2-4	指针间的转换	209
8-3	指针与函数	211
8-3-1	传递指针作为参数	214
8-3-2	返回指针	219
8-4	指针与结构	219
8-5	指针与数组	224
8-5-1	一维数组	224
8-5-2	多维数组	231
8-6	内存管理	237
第 9 章	重载 (Overload)	241
9-1	函数的重载	242
9-1-1	以参数个数区分	243
9-1-2	以参数类型区分	245
9-2	运算符的重载	249
9-2-1	重载的语法	251
9-2-2	重载一元运算符	253
9-2-3	重载二元运算符	255
第 10 章	调试	261
10-1	什么是“调试”	262
10-2	认识 Bugs	262
10-3	设置断点	264
10-3-1	调试器 (Debugger) 与断点	264
10-3-2	调试程序范例	264
10-3-3	设置断点的调试方法	265
10-3-4	检查断点变量值的变化	269
10-4	逐步执行程序	275
第 11 章	对象与类	277
11-1	类与对象的概念	278
11-1-1	类与结构	278
11-1-2	类与对象	279
11-2	类的成员函数	282
11-2-1	类函数的使用	282
11-3	构造函数	287
11-3-1	概念解析	287
11-3-2	利用构造函数初始化类变量	288
11-3-3	构造函数与参数的传递	290



11-4	对象的复制	295
11-4-1	概念解析	296
11-4-2	范例程序	296
11-5	析构函数	298
11-5-1	概念解析	298
11-5-2	范例程序	299
第 12 章	继承	303
12-1	使用继承的原因	304
12-2	基类与派生类	306
12-2-1	继承的语法	306
12-2-2	重载成员	309
12-2-3	派生类的构造函数	312
12-2-4	继承与类的转换	316
12-3	访问控制	317
12-4	多重继承	318
12-4-1	多重继承的语法	319
12-4-2	多重继承的问题	319
第 13 章	成员函数的其他特性	321
13-1	virtual 函数	322
13-1-1	改变基类函数	322
13-1-2	使用 virtual 函数	325
13-2	抽象类	328
13-2-1	概念解析	328
13-2-2	范例程序	330
13-3	friend 函数	333
13-3-1	概念解析	333
13-3-2	范例程序	333
13-4	Static 函数	336
13-4-1	Static 数据成员	337
13-4-2	Static 成员函数	339
第 14 章	文件与流	343
14-1	流类	344
14-1-1	ios 类	345
14-1-2	ostream 类	346
14-1-3	istream 类	348
14-2	流错误	349
14-2-1	状态位	349

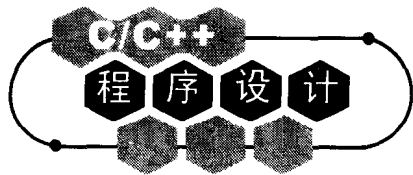
14-2-1 状态函数	350
14-3 磁盘文件 I/O	351
14-3-1 写入数据	352
14-3-2 读取数据	353
14-3-3 包含隐藏空格的字符串	353
14-3-4 检测文件结尾	354
14-3-5 关闭文件	355
14-3-6 二进制 I/O	355
14-4 对象 I/O	358
14-4-1 写入对象及读取对象	359
14-5 文件指针	361
14-6 多重对象 I/O	362
14-6-1 fstream 类	363
14-7 文件 I/O 的错误处理	365
14-7-1 错误处理	366
14-7-2 范例程序	366
14-8 使用内存	369
14-9 命令参数	372
14-10 重定向 (Redirection)	374
14-11 使用打印机	375

0

认识 C++

- 0-1 关于面向对象
- 0-2 为什么要使用面向对象
- 0-3 面向对象的优点
- 0-4 C++与 C





C 以及 C++ 语言，可以说是当今使用人数最多的程序语言。C 语言兼具有高级语言的容易学习以及低级语言强大的控制能力这两种特性，也因为这两种特性，让它取代了许多程序语言的地位，成为最多设计师爱用的程序语言。

而 C++ 则承继了 C 易学、功能强的特性，再加上面向对象的优点，让这个同样是面向对象出身的 C++ 语言，成为继 C 语言之后另一个令人不敢忽视的风云人物。

0-1 关于面向对象

面向对象 (Object-Oriented) 的技术是目前软件工业的主流，绝大多数的系统、应用程序都是采用面向对象技术所开发出来的。

面向对象的技术起源于程序设计的技巧。“对象 (Object)”、“类 (Class)” 的概念在 60 年代末期就已经出现在 Simula 程序语言中。对象与类的概念相当的简单，但是功能却非常强大。面向对象程序语言特征有：

- 数据抽象化 (Data Abstraction)
- 数据封装 (Data Encapsulation)
- 继承 (Inheritance)
- 信息传递 (Message Passing)
- 对象组合
- 多态性 (Polymorphism)
- 动态联编 (Dynamic Binding)

面向对象语言，与面向过程 (Procedure-Oriented) 语言一样，都只是为了解决问题或者开发系统、应用程序的工具。

实际生活中，对象是我们经常可以接触到的，而传统的程序设计方式却没有使用这种概念，面向对象的概念，将程序设计的领域与日常生活拉得更近，而且通过 C 语言强大的软硬件控制功能，计算机也变得更贴近人性了。

0-2 为什么要使用面向对象

传统的程序设计方式有些无法突破的阻碍，多少造成了一些使用和设计上的不便，想要更深刻了解面向对象语言的优越之处，必须先弄清楚传统的程序语言到底发生了什么问题，而面向对象语言又如何解决这些问题。

0-2-1 浅谈传统的数据处理

几十年来，我们心中都有一些概念——数据经过处理就变成了信息。计算机具有存储大量数据以及接受程序指令、快速处理数据的功能。但是对传统的程序语言来说，数据与处理却是彼此分开的，前者被动，后者主动。传统的面向过程语言，将整个世界一分为二。

数据 (Data)

描述这个世界的状态所构成的数据结构 (Data Structure)，例如人的身高、体重，或是汽

车的颜色等。

过程 (Procedures)

操作这些状态数据的步骤，例如帮汽车烤漆、改变颜色，或是减肥等。

换句话说，数据是静态、不会自行变动的。这样子的世界需要通过程序操作改变数据，进而让整个世界开始运行。将这种概念对应到程序语言时，各种数据结构的语法、函数的调用便随着而产生，但是这种想法对于真实世界而言，却是不自然的。

0-2-2 面向对象的数据处理方式

相对于面向过程的概念，面向对象则是由对象所构成的。任何一个对象都具备两个特征：行为 (Behaviors) 与状态 (States)。

面向对象语言最基本的概念就是，将数据与处理数据的函数合并成一个单元，每一个单元都称之为一个“对象”。

C++把对象中处理数据的函数称为“成员函数 (Member Function)”，每个成员函数会读取所需的数据，然后将这些数据进行处理，再将执行结果返回。这也是面向对象中唯一访问数据的途径，其他的函数将无法接触数据，数据是被隐藏的，所以可以避免意外损毁的危险；而数据与函数则被封装在一个单元中。数据封装与隐藏是面向对象语言中极为重要的概念。

如果想访问数据，唯有通过该对象的成员函数才能达成任务，其他的函数都没有这种权限，因此可以让程序的编写与调试更为容易。

每个对象通过行为的变化来改变自身的状态。也就是说，在单纯的面向对象世界中，基本上并无外力的推引操作，一切都是通过各个对象之间彼此协调而运行的。

0-3 面向对象的优点

面向对象程序语言是由几个重要的部分与概念所组成的，这些概念让面向对象成为目前最盛行的程序语言规格，例如 C++、Java、Delphi，或是即将改变成面向对象的 Visual Basic。

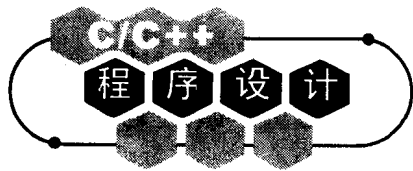
这些程序语言都具有面向对象语言的安全性以及方便性，也都成为程序设计人员最喜欢的工具。

0-3-1 对象

对象乃是面向对象语言最重要的部分，对象也是面向对象语言中构成整个程序的主要成员。

如果使用面向对象的设计原则编写程序时，不应该把重心摆在如何将一个程序分解成哪些函数，而是如何把一个程序分解成一组对象的总和。如果秉持着这种思想，您将可以发现编写程序是一件相当轻松愉快的事，因为程序语言中的对象与现实生活中的对象有着紧密的相似性。

在面向对象程序语言中，哪些东西有资格成为对象呢？其实并没有一个明确的答案，您必须自己想象才行。不过，还是有迹可寻的，像是下列生活中的对象，可以帮助思索这个问



题。

- 真实的对象：电路设计的电子组件、各个国家的经济模型。
- 计算机——用户操作界面：窗口、按钮、鼠标、键盘。
- 各种人员：雇员、经理、顾客、推销员。
- 数据的集合：字典、人事档案、库存资料。

就面向对象的角度而言，除了具备描述状态的数据外，还必须具备一些行为，也就是说，对象是一个主动的物体，我们只需要告诉对象该做什么即可，不必经过设计师加以操作即可完成指定的工作。

面向对象的概念将函数与数据合而为一，让程序中的数据与数据处理方式，与生活中的种种情形结合得更加紧密。

0-3-2 类

在真实世界中，许多对象都会具有相同的性质，例如每一个人都只是世界上所有人类的其中之一，都具有所有人类共同的特性，但是每个人又不会完全相同。

在面向对象的世界中，“人类”是一种“类（Class）”，可以看作是一个模板；而每个人，则称为这个类下的一个“实例（Instance）”。因此类可以被定义成：“类”是所有相似对象的状态变量与行为所构成的模板（Template）或原型（Prototype）。

换句话说，类像是一个制作模型的模子，利用这个模子可以做出许多相似但独立的对象。

类的使用，毋庸置疑地可以获得许多好处，像是模块化、软件芯片的制造，以及重复使用等，都是面向对象所带来的便利性。

0-3-3 继承

基本上，对象是由类所制造出来的，因此通过认识类，就可以大概知道对象长什么样子。

如果不知道什么是奔驰，只要明白它是一种汽车，马上就可以了解到奔驰是一种具有四个轮胎、长方形的车子。

在面向对象的世界中，这种概念是相同的，换句话说，类可以经过其他已知的类加以定义。举个例子，大卡车是汽车的一种，但是又与一般的汽车略有不同，所以大卡车这种类却可以经过汽车类加以定义，取用汽车类的特性，并且加以修改，就可变成大卡车类了。

在面向对象中，汽车被称作大卡车的“超类（Super Class）”，也可以称为“基类（Base Class）”；而大卡车则是汽车的“子类（Subclass）”或是“派生类（Derived class）”。这种继承关系的结构，被称为“类阶层（Class Hierarchy）”。

每一个子类基本上都可以继承其超类的特性，这些特性包含了状态数据与行为。因此大卡车也具有方向盘、煞车、轮胎等数据，拥有行走样子的行为。

不过，子类可以拥有与超类不同的地方，目的就是要让子类可以加入所需的数据与行为，比方说，大卡车可以加上 8 个轮胎，或是连接上货柜载运货品。

此外，子类可以“局部修改（Modification）”超类的行为，让这些行为更符合运行的需求，例如在大卡车的行驶方法中，可以加装马力更强的引擎，以便载运更多的物品，此时，