

高等学校21世纪教材

GAODENG XUEXIAO 21 SHIJI JIAOCAI

C++语言 基础教程

● 吕凤翥 编著



人民邮电出版社
POSTS & TELECOM PRESS

高等学校 21 世纪教材

C++语言基础教程

吕凤翥 编著

人民邮电出版社

图书在版编目 (CIP) 数据

C++语言基础教程 / 吕凤翥编著. —北京: 人民邮电出版社, 2005.11

高等学校 21 世纪教材

ISBN 7-115-13930-X

I. C... II. 吕... III. C 语言—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 122885 号

内 容 提 要

本书系统讲述 C++ 语言的基础知识、基本语法和编程方法。重点讲述 C++ 语言面向对象的重要特征: 类和对象、继承性和派生类、多态性和虚函数等重要内容。同时, 还介绍 C++ 语言对 C 语言的继承和改进。

本书文字通俗易懂, 内容全面系统, 例题丰富, 重点突出, 注重实践, 偏重应用。每章后备有大量练习题和上机题。

本书适用于大专院校理工科学生“C++ 语言程序设计”课程的教材, 还可作为自学 C++ 语言的指导书和参考书。

高等学校 21 世纪教材

C++ 语言基础教程

◆ 编 著 吕凤翥

责任编辑 张孟玮

◆ 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>

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: 27.5

字数: 660 千字

2005 年 11 月第 1 版

印数: 1—3 000 册

2005 年 11 月北京第 1 次印刷

ISBN 7-115-13930-X/TP · 4917

定价: 35.00 元

读者服务热线: (010) 67170985 印装质量热线: (010) 67129223

丛 书 前 言

当今世界,科学技术突飞猛进,知识经济已见端倪,国际竞争日趋激烈。教育在综合国力的形成中处于基础地位,国力的强弱将越来越取决于劳动者的素质,取决于各类人才的质量和数量,这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初,我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体,是进行教学的基本工具,也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此,高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代,计算机教学内容也随之更新,尤其随着教育部“高等教育面向 21 世纪教育内容与课程体系改革”计划的实施,对教材也提出的新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授,精心策划编写了这套“高等学校 21 世纪教材”。

为真正实施精品战略,组织编写好这套教材,我们在国内高校做了系统、详细的调查,对教育部制订的教育计划做了认真的研究,还对国内外已出版的教材做了理性的分析,确立了依托国家教育计划、传播先进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中,我们多次组织了由专家及高校一线教师参加的研讨会,对现有比较出色的教材的特点及优点进行了分析,博采众长,力求实现教材权威性与实用性的完美结合。

本套教材有如下特点:

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况,在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。
2. 紧密结合当前技术的新发展,在阐述理论知识的同时侧重实用性。
3. 力求在概念和原理的讲述上严格、准确、精练,理论适中,实例丰富,写作风格上深入浅出,图文并茂,便于学生学习。
4. 为适应当前高校课程种类多、课时数要压缩的教学特点,教材不仅篇幅有很大的压缩,而且均配有电子教案,以满足现代教学新特点的需要,做到易教易学。
5. 所选作者均是国内有丰富教学实践经验的知名专家、教授,所编教材具有较高的权威性。

教育的改革将不会停止,教材也将会不断推陈出新。目前本套教材即将推出,将接受广大教学第一线教师的检验。

由于我们的水平和经验有限,这批教材在编审、出版工作中还存在不少缺点和不足,希望使用本套教材的学校师生和广大读者提出批评和建议,以便改进我们的工作,使教材质量不断提高。

编 者 的 话

本书作者长期以来从事 C++ 语言程序设计课的教学工作。为了满足广大读者渴望学习和掌握 C++ 语言这一工具,更好地从事计算机编程工作和为深入学习计算机专业知识打下基础,在原有讲稿的基础上编写了此书。

本书具有下述特点。

首先,保持了原来讲稿中突出重点、注重实践、语言简明、例题精练等特点。在原稿基础上对各章节作了适当调整,增补了少量内容,针对难点和疑点,加强了解释和指点。例题更精练了,针对性更强了。更加强了对概念理解和对分析问题解决问题能力的培养。有意地加强了对增强知识培养能力的训练。

其次,体现出以人为本的思想。教学中要强调以人为本,教材中也应体现出以人为本。学习本教材时,应该是边看书边思考,不仅可以加深理解,加强记忆,还可以培养学习思考能力。本书的许多例题中都提出了思考题,本书每章后边都从多方面给出练习题,引导读者去思考,去理解,去掌握学过的内容。计算机语言作为一种工具,重要的在于理解和应用,本书尽量加强这方面的训练。

最后,注重实践是本书的又一特点。书中强调上机实践,每章后边都写有上机指导,其目的是用来引导读者上机实践。学习语言课不上机编程等于没学。书中留给读者的许多思考题都要求能够通过上机调试进行解决。同时,让读者学会如何自己提出问题并上机解决,这才是作者的最终目的,即教会读者自学语言课。读者会在学习中遇到许多问题,应学会通过上机来解决这些问题。

本教材共分 13 章。其中第 1 章至第 7 章主要复习 C 语言中讲过的一些重要内容,同时指出 C++ 语言对 C 语言的某些改进。这些改进对学习 C++ 语言很重要。第 8 章至第 12 章主要讲述 C++ 语言面向对象的特征,这是 C++ 语言中的重点,这部分内容是 C 语言中所没有的。学好这部分对学习其他面向对象语言也会有帮助。本书最后一章作为可选内容,有课时则讲,课时不够可以自学。

关于讲授 C++ 语言的学时安排,这里仅给一个参考意见。各院校可根据自己院校的情况进行调整。按照一学期讲授 16 周,不包含复习考试,每周用 4 学时讲授,2 学时上机。具体安排如下。

章 节	讲授学时数	上机学时数
第 1 章	4	2
第 2 章	4	2
第 3 章	4	2
第 4 章	4	2

章 节	讲授学时数	上机学时数
第 5 章	4	2
第 6 章	4	2
第 7 章	4	2
第 8 章	8	4
第 9 章	8	4
第 10 章	6	3
第 11 章	6	3
第 12 章	8	4
第 13 章	(机动)	

说明:

(1) C 语言基础较好的, 第 1 章至第 7 章每章可用 3 学时授课, 上机时间也可用 2 至 3 学时。

(2) 按上述学时安排, 共讲授 64 学时, 上机 32 学时。

本书所有的程序, 包括例题、练习题以及习题答案的程序都已在 Visual C++ 6.0 版本的编译系统下通过, 有关资料与电子教案放在人民邮电出版社网站 (www.ptpress.com.cn) 下载区中。

在该书的编写过程中得到许多同事和朋友的支持和帮助, 特别是吕涛、邵慧、商孟春等同志在程序录入、调试以及校对方面给予了大量的帮助, 在此一并表示感谢。

本书中难免有错漏之处, 敬请读者批评指正。谢谢您阅读这本书。

吕凤翥

2005 年 8 月

于北京大学燕北园

目 录

第1章 C++语言概述	1
1.1 面向对象的概念	1
1.1.1 面向对象方法的由来	1
1.1.2 面向对象的基本概念	2
1.2 C++语言的特点	3
1.2.1 C++语言是面向对象的程序设计语言	3
1.2.2 C++语言继承了C语言	4
1.2.3 C++语言对C语言进行了改进	4
1.3 C++程序在结构上的特点	5
1.3.1 C++程序举例	5
1.3.2 C++程序结构上的特点	8
1.4 C++程序的实现	8
1.4.1 C++程序的编辑、编译和运行	8
1.4.2 Microsoft Visual C++ 6.0 集成开发工具简介	9
1.4.3 C++程序实现举例	16
练习题1	20
上机指导1	24
第2章 变量和表达式	25
2.1 C++语言的字符集和单词	25
2.1.1 C++语言字符集	25
2.1.2 单词及其词法规则	25
2.2 数据类型和存储类	27
2.2.1 数据类型	27
2.2.2 存储类	28
2.3 常量	29
2.3.1 常量的种类	30
2.3.2 符号常量	33
2.4 变量	35
2.4.1 变量的三要素	35
2.4.2 变量的定义格式	36
2.4.3 变量的作用域	39
2.5 运算符	41

2.5.1 运算符的种类和功能	41
2.5.2 运算符的优先级和结合性	44
2.6 表达式	45
2.6.1 表达式的值和类型	45
2.6.2 表达式求值举例	46
2.7 类型转换	52
2.7.1 自动转换	52
2.7.2 强制转换	52
练习题 2	52
上机指导 2	58
第 3 章 语句和预处理	60
3.1 表达式语句和复合语句	60
3.1.1 表达式语句和空语句	60
3.1.2 复合语句和分程序	61
3.2 选择语句	61
3.2.1 条件语句	61
3.2.2 开关语句	64
3.3 循环语句	66
3.3.1 while 循环语句	66
3.3.2 do-while 循环语句	68
3.3.3 for 循环语句	70
3.3.4 多重循环	72
3.4 转向语句	76
3.4.1 goto 语句	76
3.4.2 break 语句	76
3.4.3 continue 语句	77
3.4.4 return 语句	78
3.5 类型定义语句	78
3.6 预处理功能	79
3.6.1 文件包含命令	80
3.6.2 宏定义命令	81
3.6.3 条件编译命令	82
练习题 3	84
上机指导 3	89
第 4 章 数组	91
4.1 数组的定义格式和数组元素的表示方法	91
4.1.1 数组的定义格式	91

4.1.2 数组元素的表示方法	92
4.2 数组的赋值	93
4.2.1 数组的初始化	93
4.2.2 数组的赋值	95
4.3 字符数组和字符串	98
4.3.1 字符数组的定义和赋值	98
4.3.2 字符串的输入和输出	100
4.4 数组的应用	103
4.4.1 一维数组的应用	103
4.4.2 二维数组的应用	106
练习题 4	108
上机指导 4	113
第 5 章 指针和引用	115
5.1 指针和指针的定义格式	115
5.1.1 什么是指针	115
5.1.2 指针的定义格式	116
5.2 指针的运算	117
5.2.1 指针的赋值运算和增值运算	117
5.2.2 指针的比较运算和相减运算	119
5.2.3 指针运算和地址运算	120
5.3 指针和数组	121
5.3.1 指针可表示数组元素	121
5.3.2 字符指针和字符串处理函数	124
5.3.3 指向数组的指针和指针数组	127
5.4 引用	130
5.4.1 引用和引用的创建方法	130
5.4.2 引用和指针	132
练习题 5	133
上机指导 5	138
第 6 章 函数	139
6.1 函数的定义和说明	139
6.1.1 函数的定义	139
6.1.2 函数的说明方法	141
6.2 函数的参数和返回值	142
6.2.1 函数的参数	142
6.2.2 设置函数参数的默认值	143
6.2.3 函数返回值的实现	145

6.3 函数的调用	146
6.3.1 函数的传值调用	146
6.3.2 函数的引用调用	148
6.4 指针和引用作函数参数和返回值	149
6.4.1 指针作函数参数和返回值	149
6.4.2 引用作函数参数和返回值	150
6.5 函数的嵌套调用和递归调用	153
6.5.1 函数的嵌套调用	153
6.5.2 函数的递归调用	155
6.6 重载函数和内联函数	158
6.6.1 重载函数	158
6.6.2 内联函数	161
6.7 函数的存储类	162
6.7.1 外部函数	162
6.7.2 内部函数	163
练习题 6	165
上机指导 6	170
第 7 章 结构和联合	172
7.1 结构的概念	172
7.1.1 结构类型和结构变量的定义	172
7.1.2 结构成员的表示和结构变量的赋值	174
7.2 结构与数组	177
7.2.1 数组作结构成员	177
7.2.2 结构变量作数组元素	178
7.3 结构与函数	180
7.3.1 结构变量作函数参数和返回值	180
7.3.2 指向结构变量的指针作函数参数和返回值	182
7.3.3 结构变量的引用作函数参数	185
7.4 联合的概念	186
7.4.1 联合类型和联合变量的定义	187
7.4.2 联合与结构的异同	188
7.5 联合的应用	190
练习题 7	191
上机指导 7	195
第 8 章 类和对象 (一)	197
8.1 类的定义格式和对象的定义方法	197
8.1.1 类的定义格式	197

8.1.2 对象的定义方法	200
8.1.3 对象成员的表示	201
8.2 对象的初始化	203
8.2.1 构造函数的功能、种类和特征	203
8.2.2 析构函数的功能和特点	204
8.3 数据成员的类型和成员函数的特性	209
8.3.1 类中数据成员类型的规定	209
8.3.2 成员函数的特性	209
8.4 静态成员	213
8.4.1 静态数据成员	213
8.4.2 静态成员函数	215
8.5 常成员	216
8.5.1 常数据成员	216
8.5.2 常成员函数	218
8.6 友元函数和友元类	219
8.6.1 友元函数	219
8.6.2 友元类	222
8.7 类型转换	224
8.7.1 类型的隐含转换	224
8.7.2 一般数据类型转换为类类型	224
8.7.3 类类型转换为一般数据类型	225
练习题 8	226
上机指导 8	234
第 9 章 类和对象 (二)	236
9.1 对象指针和对象引用	236
9.1.1 对象指针	236
9.1.2 this 指针	240
9.1.3 指向类的成员的指针	242
9.1.4 对象引用	244
9.2 对象数组和对象指针数组	246
9.2.1 对象数组	246
9.2.2 对象指针数组	248
9.2.3 指向对象数组的指针	249
9.3 子对象和堆对象	250
9.3.1 子对象	250
9.3.2 堆对象	252
9.4 类的作用域和对象的生存期	258
9.4.1 类的作用域	258

9.4.2 对象的生存期	258
9.4.3 局部类和嵌套类	260
9.5 类和对象的应用实例	263
练习题 9	267
上机指导 9	275
第 10 章 继承性和派生类	279
10.1 继承的概念	279
10.1.1 基类和派生类	279
10.1.2 单重继承和多重继承	280
10.1.3 派生类的定义格式	281
10.1.4 派生类成员的访问权限	282
10.2 单重继承	287
10.2.1 派生类的构造函数和析构函数	287
10.2.2 子类型和赋值兼容规则	295
10.3 多重继承	298
10.3.1 多重继承派生类的构造函数和析构函数	299
10.3.2 多重继承的二义性	302
10.4 虚基类	305
10.4.1 虚基类的概念	305
10.4.2 虚基类的应用举例	307
10.5 继承性和派生类应用举例	308
练习题 10	310
上机指导 10	319
第 11 章 多态性和虚函数	322
11.1 运算符重载	322
11.1.1 运算符重载的概念	322
11.1.2 运算符重载的两种方法	323
11.1.3 运算符重载举例	328
11.2 静态联编和动态联编	334
11.2.1 联编的概念	334
11.2.2 虚函数	338
11.2.3 动态联编	340
11.2.4 虚析构函数	343
11.3 纯虚函数和抽象类	344
11.3.1 纯虚函数	345
11.3.2 抽象类	346
11.4 应用举例	347

练习题 11	350
上机指导 11	358
第 12 章 C++语言的输入/输出操作	360
12.1 I/O 流类库概述	360
12.1.1 输入/输出流	360
12.1.2 I/O 流类库的主要功能	361
12.2 标准文件的输入/输出操作	362
12.2.1 屏幕输出操作	362
12.2.2 键盘输入操作	365
12.3 格式输出操作	369
12.3.1 使用流对象的成员函数进行格式输出	370
12.3.2 使用控制符进行格式输出	373
12.4 磁盘文件的操作	374
12.4.1 打开文件和关闭文件操作	374
12.4.2 文件的输入/输出操作	376
12.4.3 随机文件操作	380
12.4.4 文件操作的其他函数	381
12.5 字符串流输入/输出操作	384
12.5.1 字符串流的输出操作	384
12.5.2 字符串流的输入操作	385
12.6 流错误处理	386
12.6.1 错误状态字和状态函数	386
12.6.2 清除/设置错误状态位	387
练习题 12	388
上机指导 12	394
第 13 章 C++语言工具	395
13.1 模板	395
13.1.1 模板的概念	395
13.1.2 函数模板	397
13.1.3 类模板	401
13.2 异常处理	408
13.2.1 异常和异常处理	408
13.2.2 异常处理的实现	409
13.2.3 异常处理举例	412
练习题 13	416
上机指导 13	423

第 1 章 C++语言概述

本章讲述面向对象的概念和 C++语言的特点。通过 C++程序实例，分析 C++程序在结构上的特点及书写程序应注意的事项。本章还介绍使用 Microsoft Visual C++ 6.0 编译系统实现 C++程序的方法。

1.1 面向对象的概念

C++语言是一种面向对象的程序设计语言，在讲述 C++语言之前，先介绍一下有关面向对象的概念，有助于对 C++语言的理解和掌握；通过对 C++语言的学习又会进一步加深对面向对象方法的认识。

1.1.1 面向对象方法的由来

面向对象方法是人们开发软件的一种方法，这种方法的提出是软件研究人员对软件开发在认识上的一次飞跃，它是软件开发史上的一个里程碑。它标志着软件开发产业进入一个新阶段。

在面向对象方法出现之前，人们采用的是面向过程的方法。面向过程方法是一种传统的求解问题的方法。该方法将整个待解决问题按其功能划分为若干个相对独立的小问题，每个小问题还可以按其功能划分为若干个相对独立的更小的问题，依此类推，直到将所划分的小问题可以容易用程序模块实现为止。在面向过程的程序设计中，每个程序模块具有相对独立的功能，由小模块组成大模块，最后组成一个完整的程序。整个程序的功能是通过模块之间相互调用来实现的。这种面向过程的方法有很多的弊病。第一，该方法将数据和数据处理过程分离成为相互独立的实体，当数据结构一旦发生变化时，所有相关的处理过程都要进行相应的修改，因此，程序代码的可重用性较差。第二，该方法对于图形界面的应用开发起来比较困难，而图形界面越来越被人们广泛使用。第三，面向过程的程序设计中，模块之间有较大的依赖性，这对调试程序和修改程序带来一定的难度。

面向对象方法是求解问题的一种新方法，它把求解问题中客观存在的事物看做各自不同的对象，这符合人们习惯的思维方式，再把具有相同特性的一些对象归属为一个类，每个类是对该类对象的抽象描述。对象之间可以进行通信。类之间可以有继承关系，函数和运算符

可以重载，这样可以提高程序的可重用性，便于软件开发和维护。

总之，面向对象方法是计算机科学发展的要求。随着人们对信息的需求量越来越大，软件开发的规模也越来越大，对软件可靠性和代码的重用性的要求越来越高。这时，面向过程的方法使得分析结果不能直接映射待解决的问题，并且分析和设计的不一致给在编程、调试、维护等诸方面造成不便和困难。在这种情况下，面向对象方法应运而生。由于面向对象方法具有封装、继承和多态等特性，与面向过程方法相比，它较好地克服了在编程、调试和维护等方面的不便和困难，提供了代码的重用率，使得软件开发变得更为容易和方便。

1.1.2 面向对象的基本概念

面向对象是一种由对象、类、封装、继承和多态性等概念来构造系统的软件开发方法。这些新的概念描述了面向对象这种新方法。

1. 对象

对象是现实世界中客观存在的某种事物，它可以是有形的，也可以是无形的。对象是一种相对独立的实体，它具有静态特性和动态特性，通常通过一组数据来描述对象的静态特性，使用一组行为或功能来表示对象的动态特性。

对象是系统中用来描述客观事物的一个实体，它是软件系统的基本构成单位。对象是由一组属性和一组行为构成的。属性是描述对象的静态特性的数据项，行为是描述对象动态特性的操作。

2. 类

类是人们对于客观事物的高度抽象。抽象是忽略事物的非本质特性，只抓住与当前相关的特性，从而找出其共性，把具有共同特性的事物划分为一类，得到一个抽象的概念。例如，在生活中经常遇到的抽象出来的概念有桌子、房屋、汽车和足球等。

面向对象方法中的类是一种类型，它是具有相同属性和行为的对象的集合。类是具有相同属性和行为的若干对象的模板。类为属于该类的全部对象提供了抽象的描述，这种描述包括了属性和行为两大部分。类与对象的关系就像模具和铸件的关系。某个类的对象又称为该类的一个实例。

3. 封装

封装是指把对象的属性和行为结合成一个独立的单位，又称为封装体。对象的属性通常用一组数据项来表示，对象的行为又称为服务，通常用方法或函数来表示。封装体具有独立性和隐藏性。独立性表现在封装体内所包含的属性和行为形成了一个不可分割的独立单位；隐藏性表现在封装体内的有些成员在封装体外是不可见的，这部分成员被隐藏了，具有一定的安全性。一个封装体与外部联系只能通过有限的接口。

4. 继承

继承是面向对象方法提高重用性的重要措施，继承表现了特殊类与一般类之间的关系。

当特殊类包含了一般类的所有属性和行为，并且特殊类还可以有自己的属性和行为时，称作特殊类继承了一般类。特殊类又称为派生类，一般类称为基类。

继承的重要性就在于它大大地简化了对于客观事物的描述。例如，已经描述汽车这个类属性和行为，由于如小轿车是汽车类的特殊类，它具有汽车类的所有属性和行为，在描述小轿车类时，只需描述小轿车本身的属性和行为，而汽车类的属性和行为不必再重复了，因为小轿车类继承了汽车类。

5. 多态性

多态性指的是一种行为对应着多种不同的实现。在同一个类中，同一种行为可对应着不同的实现，例如，函数重载和运算符重载都属于多态性。同一种行为在一般类和它的各个特殊类中可以具有不同的实现，例如，动态联编是属于这类多态性。在一般类中说明了一种求几何图形面积的行为，这种行为不具有具体含义，因为并没有确定具体的几何图形，又定义一些特殊类，如“三角形”、“圆形”、“正方形”、“矩形”、“梯形”等，它们都继承了一般类。在不同的特殊类中都继承了一般类的“求面积”行为，可以根据具体的不同几何图形使用求面积的公式，重新定义“求面积”行为的不同实现，使之分别实现求“三角形”、“圆形”、“正方形”、“矩形”和“梯形”等面积的功能。这就是面向对象方法的重要的多态性。

1.2 C++语言的特点

C++语言是 20 世纪 80 年代初期由美国贝尔实验室的科研人员提出的，它是一种继承了 C 语言的面向对象的程序设计语言。

1.2.1 C++语言是面向对象的程序设计语言

C++语言支持面向对象的程序设计，主要表现在它支持面向对象方法中的 3 个主要特性。

1. 支持封装性

C++语言允许使用类和对象。类是支持数据封装的工具，对象是数据封装的实现。类中成员有不同的访问权限。类中的私有成员仅由该类体内的成员函数访问，因此，私有成员具有隐藏性，类体外是不可见的。类中的公有成员是类体与外界的一个接口，类体外面的函数可以访问类体中的公有成员。类中还有一种保护成员，它具有公有成员和私有成员的双重特性，它具体使用在类的继承中。

2. 支持继承性

C++语言支持面向对象方法中的继承性，它不仅支持单重继承，而且支持多重继承。继承性给 C++语言编程带来了方便，增强了程序的扩展性和可重用性，提高了软件开发的效率。继承是两个类之间的关系，基类和派生类是继承中的重要概念。派生类继承了基类中的所有

成员，并且还可以定义自身的新成员，继承实现了抽象和共享的机制。继承和封装是衡量一种语言是否是面向对象的程序设计语言的两个重要标准。C++语言支持封装，又支持继承，因此，可以断定 C++语言是面向对象的程序设计语言。

3. 支持多态性

多态性是在继承性基础上的面向对象方法中的重要特性之一。C++语言支持多态性主要表现为如下两个方面。

(1) 支持函数重载和运算符重载。重载是指一个函数名可以有多种实现，即同一个行为对应不同实现，这便是多态性。

(2) 支持动态联编。动态联编反映了基类和派生类中同名函数的多态性。动态联编是在公有继承的前提下，通过虚函数来实现的。动态联编虽然没有静态联编运行效率高，但它可以通过高度抽象，提高程序的灵活性和扩充性。

1.2.2 C++语言继承了 C 语言

C++语言与 C 语言兼容，C 语言是 C++语言的一个子集。C 语言的词法、语法和其他规则都可以用到 C++语言中。例如，C 语言中的类型、运算符和表达式在 C++语言中都可以使用；C 语言中的语句也是 C++语言的语句；C 语言中的函数定义和调用也适用于 C++语言，只是 C++语言对此稍有改进和扩充；C 语言中的预处理命令也可用于 C++语言，只是宏定义命令在 C++语言中较少使用；C 语言中的构造类型，如数组、结构和联合在 C++语言中也是适用的，只不过 C++语言中较多是使用类类型；C 语言中的指针在 C++语言中也可使用，但是 C++语言中引入了引用概念，在某些情况下减少了指针的使用；还有 C 语言中作用域的规则，存储类的规定等在 C++语言中也都适用。

由于 C++语言继承了 C 语言，使得已经掌握了 C 语言的人们学习 C++语言比较容易，这也是 C++语言得以广泛使用的原因之一。已经学会了 C 语言的人们学习 C++语言时，要做到一种思维方式的转变，即从面向过程的思维方式转变到面向对象的思维方式，没有这个转变是学不好 C++语言的。

由于 C++语言继承了 C 语言，因此，C++语言仍旧具有 C 语言的简练明了的风格，同时还不得不保留某些 C 语言的面向过程的特性。所以，有人说 C++语言是一种不完全的面向对象的程序设计语言。在 C++程序中，允许类体外的函数存在，这便是保留面向过程的特征。

1.2.3 C++语言对 C 语言进行了改进

C++语言虽然保留了 C 语言的风格和特点，但又针对 C 语言的某些不足做了较大的改进。改进后的 C++语言与 C 语言相比，在数据类型方面更加严格了，使用更加方便了。

下面简单扼要地列举一些 C++语言对 C 语言的改进内容，更详细的讲述可参见本书第 2 章至第 7 章。

(1) C++语言中规定所有函数定义时必须指出数据类型，不允许默认数据类型。无返回值的函数使用 `void` 进行说明，返回值为整型的函数使用 `int` 进行说明。

(2) C++语言规定函数说明必须使用原型说明，不得用简单说明。