

开发者说系列丛书

C++ STL

程序员开发指南

彭木根 王淑凌 编著

- 用较大的篇幅对C++语言编程核心技术、C++关键库类和 C++STL编程技术进行了深入地剖析和源码解析。
- C++领域内的一本权威著作，是目前市场上惟一采用实例的形式全面讲述STL技术的书籍。
- 融合并提炼了许多程序员多年开发C++程序积累下来的成熟经验。
- 书中的内容、知识点、实例既适合课堂教学，也适合读者自学。

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

C++STL 程序员开发指南

彭木根
王淑凌
编著

本书附光盘 / 张



燕山大学图书馆藏书

TP 312C / 159

中国铁道出版社



0654081

-85

517
102

(京)新登字 063 号

内 容 简 介

本书通过对大量程序实例的分析,深入浅出地讲解了 C++ STL 高级编程技术。全书首先介绍了 C++语言的基本知识、C++语言编程核心技术和 C++关键库类,然后逐步过渡到 C++ STL 编程技术,用较大的篇幅对它们进行了深入的剖析和源码解析。

本书由大量的源程序实例组成,融合并提炼了许多人多年开发 C++程序积累下来的成熟经验,意在展现深奥及抽象的 C++编程技术,特别是令人望而生畏的强大的 STL 技术。

本书是 C++领域内一本权威的著作,是目前市场上惟一一本采用实例的形式全面讲述 STL 技术的书籍。书中的内容、知识点、实例既适合课堂教学,又适合读者自学。无论是高等院校计算机及相关专业的学生,还是计算机业界的从业人员,以及广大的计算机爱好者,都可以从本书中获益。

图书在版编目(CIP)数据

C++ STL 程序员开发指南/彭木根,王淑凌编著.北京:中国铁道出版社,2003.3

(开发者说)

ISBN 7-113-05164-2

I. C… II. ①彭… ②王… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 017707 号

书 名: C++STL 程序员开发指南

作 者: 彭木根 王淑凌

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: 严晓舟 郭毅鹏

责任编辑: 苏 茜 彭立辉

封面设计: 孙天昭

印 刷: 河北省遵化市胶印厂

开 本: 787×1092 1/16 印张: 32 字数: 754 千

版 本: 2003 年 4 月第 1 版 2003 年 4 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-05164-2/TP·900

定 价: 56.00 元

版权所有 侵权必究

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。

前言

在 C++ 的最新发展过程中, 新增了模板这个新特性, 通过使用模板, 程序具备更好的代码重用性能。STL 是在加利福尼亚州的惠普实验室中开发的一系列软件的统称, 开发者主要为 Alexander Stepanov、David R Musser 和 MengLee 三位大师, 其中 Stepanov 更被誉为“STL 之父”。在 1994 年 7 月, 美国国家标准研究所(ANSI) 通过投票决定, 将 STL 纳入 C++ 标准, 使之成为 C++ 库的重要组成部分。目前我们所讲的 C++ 标准是经过多次修改的结果, 最近一次的修改在 1997 年 11 月, 其官方名称为 ISO14882, 其最重要的修改来自于把模板引入到标准库, 并且使用模板类代替 C++ 中传统的类格式, 这完全符合 STL 的概念。

1998 年, C++ 标准正式通过, 并且将在接下来的一段时间中保持一种稳定的状态。它既是 ANSI 标准, 也是 ISO 标准, 也就是说, 它既是美国的国家标准, 也是一种全球通用的标准。作为标准化进程的一部分, 整个标准 C++ 语言及库第一次完整地描述在一起。一个相对较晚加入这个标准化进程的事件是在 C++ 标准的草案中接纳 STL 为 C++ 标准的一部分。同样, 在 1994 年 6 月被标准化委员会接纳为 C++ 标准草案的一部分之前, 对 STL 的早期描述(至少早于 S&L95) 仅限于一个相对狭窄的范围内。在被接纳的过程中, STL 本身又被重新组织并且在几个重要的环节做了修改。现在, 在标准 C++ 中的 STL 已经不再是由惠普实验室开发出来的那套软件开发包, 同时它与那些由不同的代理提供的强化版本也不同。

STL 能够给编程人员带来很多的好处, 比如熟悉了 C++ STL 编程思想后, 很多用传统的 C++ 编写的代码实际上用几句 STL 编码就可以实现。通过调用一两个算法模板, 就可以得到一些幽雅但绝对高效的代码, 所以 STL 技术出现以来就一直非常流行, 而且发展劲头强劲。近几年, 中国也开始旋起了一股“STL”热, 但摆在 STL 爱好者面前的是没有相应的资料, 特别是适合中国国情的技术资料几乎没有。另外, 要掌握和精通 C++ STL 技术, 必须精通 C++ 高级编程技术, 而从 C++ 过渡到 C++ STL, 对很多爱好者来说是一片茫然。

本书从程序员的角度出发, 首先讲述了 C++ 高级编程技术, 然后讲述了 C++ 标准库, 提到了 C++ 的模板技术, 自然而然的提出了 C++ STL 技术。之后是重点讲述 STL 的原理、组成和结构。通过大量的实例阐述了 STL 技术的使用。之所以称本书为《C++ STL 程序员宝典》, 主要在于本书的实例可以被读者直接拷贝使用, 里面的编程方法和技巧可以直接被读者所采纳。

本书的编写由彭木根、王淑凌、梁永明、马晓军、聂世杰、杨昆、王秀明、李晓明、贺兵、任兴安等同志完成, 并得到郭毅鹏的大力支持和帮助。另外, 陈贤淑、陈晓娟、廖康良等同志参与了本书的编排工作, 在此一并表示感谢! 由于时间紧迫以及作者水平有限, 书中难免有不足之处, 恳请读者批评指正。我们也会在适当时间进行修订和补充, 并发布在天勤网站: <http://www.tqbooks.net> “图书修订”栏目中。

编者

2003 年 3 月

0304

目 录

第一篇 预备知识

第 1 章 C++编程技术	3
1-1 C++与 C 语言的区别.....	4
1-1-1 文件扩展名的改变.....	4
1-1-2 简化输入/输出手段	5
1-1-3 数据类型声明的改变.....	5
1-1-4 动态内存分配运算符的使用.....	6
1-1-5 引用 (References) 类型.....	8
1-1-6 const 语义的扩展	9
1-1-7 指针声明类型与对象类型相一致.....	13
1-1-8 int 与 char 不再等价.....	13
1-1-9 结构数据类型的变化	13
1-1-10 数组和指针技术的不同.....	14
1-2 C++存储技术.....	15
1-2-1 C++存储类型	15
1-2-2 C++存取修饰符	16
1-2-3 C++对象的生存期	17
1-3 C++函数技术.....	19
1-3-1 类的构造函数、析构函数与赋值函数.....	19
1-3-2 在派生类中实现类的基本函数	29
1-3-3 内联函数技术.....	30
1-3-4 友元函数技术.....	31
1-4 C++面向对象机制的实现.....	33
1-4-1 类的继承技术.....	33
1-4-2 函数重载技术.....	37
1-4-3 运算符重载技术.....	38
1-4-4 纯虚函数和抽象类技术.....	40
1-5 小结	42
第 2 章 C++标准库技术	43
2-1 C++标准库简介.....	44
2-1-1 I/O 流技术.....	46
2-1-2 string 类.....	48
2-1-3 标准异常类.....	48
2-1-4 标准模板库类.....	49

2-2 C++输入/输出流技术	50
2-2-1 C++语言输入/输出流概述	50
2-2-2 输入/输出格式控制	50
2-2-3 特殊输入/输出流格式的设定	55
2-2-4 自定义的流操作符	60
2-2-5 根本不用于标准流运算符间的流字符串读/写函数	61
2-2-6 标准输入/输出流的操作符的重载	62
2-2-7 C++文件输入/输出流	63
2-3 C++字符串技术	75
2-3-1 字符串类基本操作	75
2-3-2 复杂字符串实例	89
2-4 标准异常类	92
2-4-1 绝对终止机制	92
2-4-2 非局部 Goto 机制	95
2-4-3 signals 机制	95
2-4-4 C++异常处理机制	97
2-4-5 类的异常处理	99
2-5 小结	100

第二篇 C++ STL 技术原理和组成

第3章 STL 技术原理	103
3-1 模板概述	104
3-1-1 Smalltalk 方法	104
3-1-2 模板方法	105
3-1-3 模板参数	106
3-1-4 关键字 typename 的使用	107
3-2 函数模板	108
3-2-1 函数模板基础	108
3-2-2 函数的定制	110
3-2-3 函数模板实例	112
3-3 类模板	115
3-3-1 类模板定义	116
3-3-2 类模板使用	118
3-3-3 类模板中的友元	118
3-3-4 模板程序设计举例	119
3-4 模板安全	122
3-4-1 Class 类型的参数	123
3-4-2 包容安全	124
3-4-3 默认构造函数	124

3-4-4	operator new	125
3-4-5	Destructor	125
3-4-6	其他.....	126
3-5	模板的特殊性	129
3-5-1	一个特殊化的例子.....	129
3-5-2	指针特殊化.....	130
3-6	模板实例——list 容器类设计	132
3-7	小结	135
第 4 章	STL 技术概述	137
4-1	STL 简介	138
4-1-1	什么是 STL.....	138
4-1-2	STL 的发展.....	139
4-1-3	STL 的使用和实现——确定无益后删除	140
4-1-4	命名空间技术.....	142
4-2	STL 基本结构	143
4-2-1	容器 (Containers)	144
4-2-2	算法 (Algorithms)	148
4-2-3	迭代器 (Iterators)	150
4-2-4	函数对象 (Function Object)	153
4-2-5	其他部件.....	154
4-3	STL 编程概述	154
4-3-1	传统 C++ 编程技术.....	155
4-3-2	STL 编程技术.....	156
4-4	STL 头文件和编译器	161
4-5	STL 与 MFC 比较	162
4-5-1	MFC 的缺陷	163
4-5-2	STL 与 MFC 指针技术的差别.....	163
4-5-3	STL 技术与 MFC 的互补.....	165
4-5-4	STL 与 MFC 的类差别.....	167
4-5-5	STL 与 MFC 的互相转换.....	174
4-6	STL 编程关键	176
4-5-1	STL 容器技术.....	176
4-5-2	C++ 和 STL 技术.....	178
4-7	小结	184
第三篇 C++ STL 容器编程技术		
第 5 章	STL 容器技术总述	189
5-1	容器技术概述	190
5-1-1	容器介绍	193

5-1-2	Forward 容器	194
5-1-3	Reversible 容器	195
5-1-4	Sequence	195
5-1-5	Associative 容器.....	196
5-2	vector 技术.....	196
5-2-1	vector 头文件	196
5-2-2	vector 对象	197
5-2-3	vector 实例	199
5-3	deque 技术	200
5-3-1	deque 头文件.....	201
5-3-2	deque 对象.....	201
5-3-3	deque 实例.....	203
5-4	list 技术	204
5-4-1	list 头文件.....	205
5-4-2	list 对象.....	205
5-4-3	list 实例.....	208
5-5	stack 技术.....	209
5-5-1	stack 头文件	209
5-5-2	stack 对象	210
5-5-3	stack 实例	211
5-6	queue 技术	213
5-6-1	queue 头文件.....	213
5-6-2	queue 对象.....	213
5-6-3	queue 实例.....	214
5-7	priority_queue 技术	216
5-7-1	priority_queue 头文件.....	217
5-7-2	priority_queue 对象.....	217
5-7-3	priority_queue 实例.....	218
5-8	slist 技术.....	220
5-9	关联式容器	221
5-9-1	set 介绍	221
5-9-2	multiset 介绍	224
5-9-3	map 介绍.....	228
5-9-4	multimap 介绍.....	233
5-9-5	实例详解.....	237
5-10	小结	238
第 6 章 vector 技术编程详解		239
6-1	vector 编程入门	240
6-1-1	定义 vector	240

6-1-2	vector 初始化	242
6-1-3	vector 大小统计	243
6-2	vector 基本使用	244
6-2-1	判断 vector 是否空	244
6-2-2	使用循环遍历成员	246
6-2-3	使用迭代器	247
6-2-4	使用算法	248
6-3	vector 高级编程技术	251
6-3-1	vector 对象的查找	251
6-3-2	vector 对象的搜索	252
6-3-3	vector 字符串处理	253
6-3-4	vector 的排序	255
6-3-5	vector 元素增加	256
6-3-6	vector 元素删除	257
6-3-7	vector 对象交换	260
6-4	vector 程序综合实例分析	261
6-5	小结	264
第 7 章	deque 技术编程详解	265
7-1	deque 编程入门	266
7-1-1	deque 的定义	266
7-1-2	deque 赋值	268
7-1-3	deque 大小度量函数	270
7-1-4	返回函数	271
7-2	deque 编程深入	276
7-2-1	判断容器是否为空	276
7-2-2	deque 访问	278
7-2-3	deque 重置技术	279
7-2-4	容器内容交换	280
7-3	deque 插入和删除技术	282
7-3-1	insert 操作	283
7-3-2	erase 操作	284
7-3-3	clear 操作	285
7-4	deque 模板函数详解	286
7-4-1	operator[]	286
7-4-2	operator==	288
7-4-3	operator<	288
7-4-4	operator!=	289
7-4-5	operator<=	290
7-4-6	operator>	290

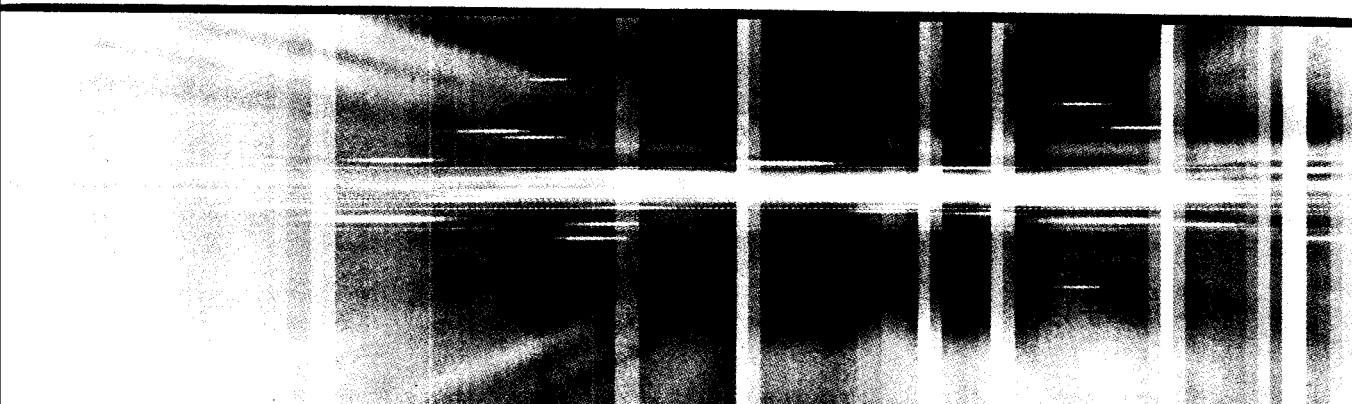
7-4-7	operator>=	291
7-5	deque 实例详解	292
7-6	小结	295
第 8 章	list 技术编程详解	297
8-1	list 编程入门	298
8-1-1	list 的定义	298
8-1-2	list 赋值	301
8-1-3	list 大小度量函数	306
8-1-4	返回函数	309
8-2	list 编程详解	313
8-2-1	判断容器是否为空	313
8-2-2	list 访问	314
8-2-3	list 重置技术	315
8-2-4	容器内容交换	317
8-3	list 插入和删除技术	319
8-3-1	insert 操作	319
8-3-2	erase 操作	321
8-3-3	clear 操作	323
8-4	list 模板函数详解	323
8-4-1	operator==	324
8-4-2	operator<	324
8-4-3	operator!=	325
8-4-4	operator<=	326
8-4-5	operator>	327
8-4-6	operator>=	327
8-5	list 特殊函数	328
8-5-1	merge()函数的使用	328
8-5-2	remove()	330
8-5-3	remove_if()	331
8-5-4	sort()	332
8-5-5	splice()	333
8-5-6	unique()	335
8-6	list 实例详解	336
8-7	小结	342
第 9 章	set 和 multiset 技术编程详解	343
9-1	set 和 multiset 定义和创建	344
9-1-1	set 类模板简介	345
9-1-2	multiset 类模板简介	347
9-2	set 和 multiset 编程基础	349

9-2-1	begin 函数	349
9-2-2	end 函数	351
9-2-3	rbegin 函数	352
9-2-4	rend 函数	354
9-2-5	判断空函数	357
9-2-6	计算大小函数	358
9-2-7	元素的插入	360
9-2-8	元素的删除操作	362
9-3	set 和 multiset 深入编程	366
9-3-1	count 函数	366
9-3-2	元素的查找	368
9-3-3	上下限迭代器的返回	370
9-3-4	元素的随机访问	371
9-3-5	元素大小比较	375
9-3-6	获取内存分配器	379
9-4	set 和 multiset 编程实例	381
9-5	小结	387
第 10 章	map 和 multimap 技术编程详解	389
10-1	map 和 multimap 定义和使用	390
10-1-1	map 类模板简介	391
10-1-2	multimap 类模板简介	393
10-2	map 和 multimap 编程基础	395
10-2-1	begin 函数	395
10-2-2	end 函数	397
10-2-3	rbegin 函数	399
10-2-4	rend 函数	401
10-2-5	判断空函数	404
10-2-6	计算大小函数	405
10-2-7	元素的插入	408
10-2-8	元素的删除操作	410
10-2-9	元素的交换	414
10-3	map 和 multimap 深入编程	417
10-3-1	count 函数	417
10-3-2	元素的查找	418
10-3-3	元素相等时上下限迭代器的返回	420
10-3-4	元素的随机访问	423
10-3-5	元素大小比较	427
10-3-6	获取内存分配器	430
10-4	编程实例	433

10-5 小结	440
---------------	-----

第四篇 C++算法技术

第 11 章 通用算法技术	443
11-1 通用算法技术简介	444
11-2 非修正序列算法	448
11-2-1 查找容器中相同的相邻元素	448
11-2-2 容器中相同元素统计	449
11-2-3 容器对象变量比较	450
11-2-4 元素查找	451
11-2-5 特定的循环操作	454
11-2-6 不相等元素查找	456
11-2-7 采用 search 查找函数	458
11-3 修正序列算法	459
11-3-1 元素复制	460
11-3-2 赋值操作	461
11-3-3 通过函数进行元素的赋值	462
11-3-4 容器拆分技术	463
11-3-5 重新随机分布	464
11-3-6 元素删除	466
11-3-7 元素替换	467
11-3-8 元素的旋转	467
11-3-9 元素颠倒算法	469
11-3-10 元素交换算法	470
11-3-11 容器运算技术	472
11-3-12 删除容器中重复元素	473
11-4 排序算法	475
11-4-1 排序算法	475
11-4-2 排序元素的查找	478
11-4-3 字典式比较	483
11-4-4 极值元素求解	484
11-4-5 合并排序算法	489
11-4-6 拆分排序	492
11-4-7 堆栈操作技术	493
11-5 数值算法	495
11-5-1 元素求和	495
11-5-2 元素内积	498
11-5-3 序列相邻差	499
11-6 小结	500

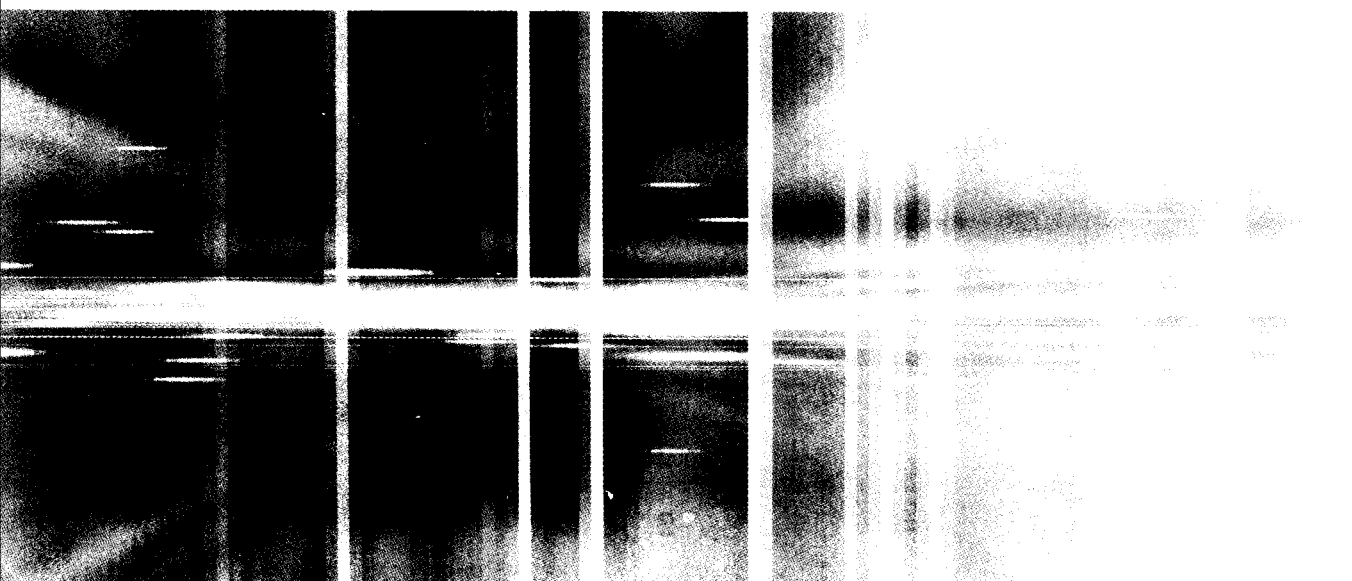


第一篇

预备知识

第 1 章 C++ 程序设计

第 2 章 C++ 标准库技术



本篇导读

在着手学习和研究 C++ STL 高级编程技术之前，需要您能够熟练地运用 C++ 编程并分析程序，这些基本的技能通常是从 C++ 课程以及其他分散的课程中学到的。本书的第一部分旨在帮助你回顾一下这些知识点，包括 C++ 的一些基本特性以及 C++ 标准类库。

第 1 章我们将回顾 C++ 的一些特性。因为不是针对 C++ 新手，因此没有介绍诸如赋值语句、if 语句和循环语句（如 for 和 while）等基本结构和基本语法，而是主要介绍一些可能已经被你忽略或忘记的 C++ 特性，如类的共享成员、保护成员和私有成员；类的继承和多态，友元和纯基类等技术。

第 2 章主要讲述了 C++ 标准库技术。由于 C++ STL 是 C++ 标准库技术中的一部分，全面了解 C++ 标准库技术是理解和掌握 C++ STL 技术的基础。本章首先介绍了什么是 C++ 标准库，然后介绍了输入/输出流，字符串类以及标准异常类等机制。至于标准模板库类技术（C++ STL），将在下面的部分集中介绍。

Chapter

1

C++ 编程技术

本章主要内容

- ✎ C++与 C 语言区别
- ✎ C++存储技术
- ✎ C++函数技术
- ✎ C++面向对象机制实现

C++语言是在基于 C 语言的语法基础上融入了 Simula67、Algol68 和 Ada 等语言中许多独特的语法特点之后而形成的全新的语言体系。但由于 C++语言的语法结构与 C 语言的语法结构几乎一致且多数编译器供应商所提供的编译器即可编译 C 语言，同时也可编译 C++语言而且还允许这两种语言混编，因而在很大程度上使许多读者误以为 C++语言是 C 语言的改进型语言。另外在事实上，C++语言也的确可以沿用 C 语言所支持传统的面向过程的分析和设计方法编制出过程化的程序，从而使更多的人对其产生了上面的误解。正是由于这些问题的存在，那些打算自学 C++语言的读者便面临着更大的误入歧途的可能性。至于将 C++语言和 C 语言融入一个编译器来进行编译处理主要还是由于存在商业上需求的缘故。

在所有的 C++语言的最基本的概念中，类（class）便是核心。读者如果已系统地学习过面向对象的分析与程序设计方法，相信学习本章的内容将不会有很大的困难。在掌握了本章论述的 C++语言的最基本的概念之后，再回头重温面向对象的程序设计方法便会更深刻地体会其精深之处了。本章主要介绍 C++的关键语法，分析 C++的许多内核技术。

1-1 C++与 C 语言的区别

C 是一种简单的语言。它真正提供的只有宏、指针、结构、数组和函数。不管什么问题，C 都是依靠宏、指针、结构、数组和函数来解决。而 C++不是这样，宏、指针、结构、数组和函数当然还存在，此外还有私有和保护型成员、函数重载、缺省参数、构造和析构函数、自定义操作符、内联函数、引用、友元、模板、异常、命名空间等。

对每个人来说，习惯 C++需要一些时间，对于已经熟悉 C 的程序员来说，这个过程尤其令人苦恼。因为 C 是 C++的子集，所有的 C 的技术都可以继续使用，但很多用起来又不太合适。正是由于 C++语言的语法是基于 C 语言，并考虑到大多数学习 C++语言的读者应当系统地掌握传统的软件工程所述的面向过程的分析与设计方法和 C 语言程序设计等，并且完全掌握面向对象编程的思想，所以本节是以此为起点来讲述 C++语言的，在同 C 语言进行比较的基础上来论述 C++语言的语法。

1-1-1 文件扩展名的改变

为了使编译器能够区别是 C 语言还是 C++语言，C++语言体系规定用“cpp”（意即 C Plus-Plus）做为 C++语言源文件的扩展名以区别于 C 语言用的“.c”文件扩展名。虽然仅差两个字母，但编译时的处理却相差甚远。后面将不再重复 C 语言已有的内容，所以凡是讲到的多属于 C++语言特有的语法描述的内容则只能书写在以“cpp”为扩展名的文件中（即按 C++语法处理）。反过来按 C 语言语法书写的程序却大多可以使用“cpp”的文件扩展名并用 C++编译器处理。但是也有个别程序易发生一些不符合预期要求的问题（如将在后面讲的重载）。所以为了避免混乱，建议还是按各自语法规则书写。“cpp”的文件扩展名与操作系统无关。与 C++语言源文件相关的头文件扩展名一般仍用“h”，但有些操作系统也有规定使用“hpp”充当头文件扩展名的。

1-1-2 简化输入/输出手段

在C语言中,输入/输出本是依靠函数来实现的(如标准输入/输出用的 `scanf`、`printf` 等)。通常只要在程序的最前端放置对应的头文件声明“`#include<stdio.h>`”就可以引用这类函数。由于此种函数的引用语句书写起来比较冗长,为了简化起见C++语言又另外定义保留字和运算符来替代C语言中对标准输入/输出函数的引用。C++语言的保留字为:

```
cout<<"输出内容"<<...; /*为标准输出算符(默认为显示器)*/
```

```
cin>>"输入内容">>...; /*为标准输入算符(默认为键盘)*/
```

支持此二算符的内部函数体在一个名为“`iostream.h`”(即标准 I/O 流)的头文件中予以声明,所以应用时一定要将其放置对应的头文件声明部位。

【例程 1-1 简化输入/输出手段实例】

```
//文件名: CHAPTER2-1.cpp
#include<iostream.h>
void main()
{
    char name[10];
    int age;
    cout<<"please input your name:";
    cin>>name;
    cout<<"How old are you:";
    cin>>age;
    cout<<"name is "<<name<<"\n";
    cout<<"age is "<<age<<"\n";
}
```

将实例中的标准输入/输出语法与C语言的 `printf` 和 `scanf` 函数的书写语法对比后可以很明显的看出此种标准的输入、输出语法的书写大大简化了函数格式的描述,而由C++语言编译器按被操作数的类型和具体内容代为选定默认格式。实际上此种方法也有其一整套控制格式的手段。运行时在屏幕输出如下:

```
please input your name: peng
How old are you: 20
name is peng
age is 20
```

1-1-3 数据类型声明的改变

C++语言除了新增的类数据(即 `class`)类型以外,还继承了C语言所支持的所有数据类型。但在数据类型的声明上作了两种较大的改变,一是允许数据类型的声明语句可以出现在程序的任何位置;二是允许直接使用结构体名定义实体。

【例程 1-2 数据类型声明实例】

```
//文件名: CHAPTER2-2.cpp
#include <iostream.h>
void main()
```