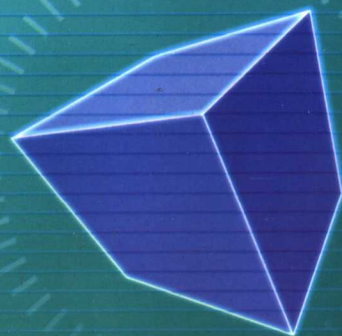
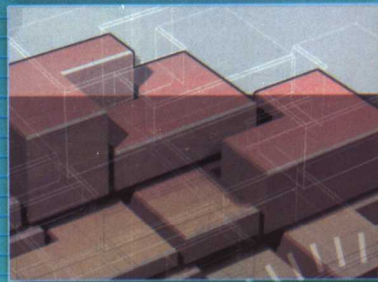
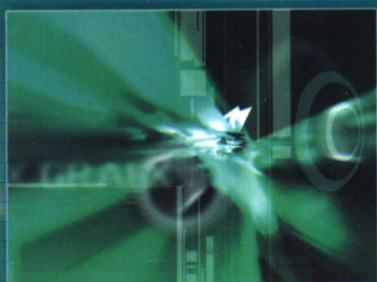




普通高等教育“十一五”规划教材
高等院校计算机技术系列教材

计算机高级语言 程序设计 (C++)

洪 洲 蔡木生 罗 林 编著



冶金工业出版社

普通高等教育“十一五”规划教材
高等院校计算机技术系列教材

计算机高级语言程序设计 (C++)

洪 洲 蔡木生 罗 林 编著

北 京

冶金工业出版社

内 容 简 介

本书是根据普通高等教育“十一五”规划教材的指导精神而编写的。

本书用浅显的语言,通过大量的实例,全面系统地介绍了高级语言程序设计中 C++ 语言的语法、数组、函数与重载、类、封装与继承和 I/O 流等基本概念,并运用这些基本概念进行面向过程及面向对象的程序设计方法讲解。学好 C++,很容易触类旁通学习其他的高级语言, C++ 是计算机高级语言中的经典,也是通向软件开发应用的桥梁。

本书是作者总结多年教学经验并结合自身的实际项目工程经验编写而成的,适合用作高等院校计算机专业和非计算机专业的程序设计基础教材,也可供 C++ 程序员和计算机软件技术人员作为程序设计的参考书。

图书在版编目 (CIP) 数据

计算机高级语言程序设计: C++ / 洪洲, 蔡木生, 罗林
编著. —北京: 冶金工业出版社, 2007.6

普通高等教育“十一五”规划教材. 高等院校计算机
技术系列教材

ISBN 978-7-5024-4302-3

I. 计… II. ①洪…②蔡…③罗… 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 (2007) 第 076306 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

ISBN 978-7-5024-4302-3

广州锦昌印务有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2007 年 6 月第 1 版第 1 次印刷

787mm × 1092mm 1/16; 23 印张; 531 千字; 358 页

39.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于本书

本书是根据普通高等教育“十一五”规划教材的指导精神而编写的。

在各种面向对象的高级程序设计语言中，C++最受人青睐。一方面，C++是C的超集，是C语言的延伸，而C已拥有了最广泛的用户；另一方面，它既可进行面向过程的程序设计，也可进行面向对象程序设计，是生成代码效率最高的语言之一。而且C++实现了类的封装、继承及多态，使得其代码更容易维护和具有高度可重用性。

学习C++不仅可以深入理解面向对象程序设计的一些基本概念，而且非常有助于进一步学习其他的计算机语言。

本书着重介绍VC++开发环境下编程工具的使用、C++基本理论思想和基本编程技能，符合20%80%法则，介绍计算机20%的功能，做80%的事。

二、本书结构

本书共有16章，可分为三大部分，具体内容安排如下：

第一部分为C++语言基础。

第1章：C++语言概述。主要介绍C++语言的基本知识、C++语言的语法规则与书写格式、VC++ 2005集成开发环境简介等内容。

第2章：数据类型、运算符与表达式。主要介绍C++基本数据类型、常量、变量、C++运算符与表达式等内容。

第3章：C++程序的结构。主要介绍编译预处理功能、语句与程序结构、数据的输入与输出等内容。

第4章：程序流程控制结构。主要介绍顺序结构、选择控制结构、循环控制结构，以及break语句与continue语句等内容。

第5章：数组与字符串。主要介绍一维数组、二维数组、字符数组等内容。

第6章：指针与引用。主要介绍指针的基本概念、指针变量、指针运算、指针与数组、动态内存分配（new与delete）、引用等内容。

第7章：函数。主要介绍函数定义的基本知识、函数调用、变量及作用域、内部函数和外部函数等内容。

第8章：函数的高级特性与应用。主要介绍函数与数组、函数与指针、内联函数、函数的缺省参数、函数重载、函数的嵌套调用等内容。

第9章：自定义数据类型。主要介绍结构的基本知识、枚举类型、用typedef声明类型等内容。

第二部分为面向对象的程序设计方法。

第10章：类与对象。主要介绍类的基本知识、对象、this指针、构造函数、对象与指针、常对象与常成员等内容。

第 11 章：静态成员与友元。主要介绍静态成员基本知识、友元的基本知识等内容。

第 12 章：运算符重载。主要介绍运算符重载的基本方法、运算符重载的一般规则、常用运算符的重载、类型转换等内容。

第 13 章：继承与派生。主要介绍继承与派生的基本知识、派生类的构成、类的继承方式、单一继承下的构造函数与析构函数、多重继承、虚基类的基本知识等内容。

第 14 章：虚函数与多态性。主要介绍静态联编和动态联编的概念、虚函数、虚析构函数、纯虚函数和抽象类、多态性等内容。

第 15 章：输入输出流与文件操作。主要介绍输入输出流的概念、标准输出流、标准输入流、文件流与文件操作等内容。

第三部分为上机实验。

第 16 章：上机实验。主要是针对前面每一章的上机实验。

三、本书特点

本书通过循序渐进，由浅到深的方式，结合大量典型例子，以简单明了、通俗易懂的语言介绍了 C++ 高级语言程序设计的基本理论和实践应用。注重内容的先进性与时代的同步性，有机地融合和引入了多种流行、先进的技术和理念，引导学生掌握和应用最新的先进技术与成果，激发学生的学习热情和兴趣，深入学习、掌握和应用相关技术。

四、本书适用对象

本书适合用作高等院校计算机专业和非计算机专业的程序设计基础教材，也可供 C++ 程序员和计算机软件技术人员作为程序设计的参考书。希望本书能为读者学习 C++ 及计算机高级语言程序设计提供强有力的帮助，这也是我们为之不断努力的目标。

本书由洪洲、蔡木生、罗林共同编写，第 1 章至第 5 章由洪洲编写，第 6 章至第 10 章由罗林编写，第 11 章至第 15 章由蔡木生编写，全书由洪洲统稿。

由于编写时间仓促，编者水平有限，书中疏漏和不足之处在所难免，敬请有关专家、学者与读者们谅解和指正。联系方法如下：

电子邮箱：service@cnbook.net

网址：www.cnbook.net

本书电子教案、源代码和习题参考答案可从该网站下载，此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者

2007 年 5 月

目 录

第 1 章 C++语言概述1

1.1 计算机高级语言发展概况1

1.1.1 计算机高级语言产生的历史 背景1

1.1.2 计算机高级语言的现状与发展1

1.1.3 C++语言的发展2

1.2 面向对象程序设计方法概述3

1.2.1 传统的结构化程序设计3

1.2.2 面向对象的程序设计3

1.3 C++语言的语法规则与书写格式4

1.3.1 标识符与关键字4

1.3.2 C++语言的应用程序组成5

1.3.3 C++的语法规则与书写格式6

1.3.4 简单的 C++语言程序范例6

1.4 程序开发过程8

1.5 VC++ 2005 集成开发环境简介8

1.5.1 VC++ 2005 常用功能介绍9

1.5.2 VC++ 2005 控制台应用程序 简介10

小结12

综合练习一12

一、选择题12

二、填空题13

三、思考题13

四、上机操作题13

第 2 章 数据类型、运算符与表达式15

2.1 C++基本数据类型15

2.1.1 整数类型15

2.1.2 浮点类型17

2.2 常量17

2.2.1 数值常量17

2.2.2 字符串常量18

2.2.3 常量定义19

2.3 变量20

2.3.1 变量定义20

2.3.2 变量赋初值与初始化20

2.4 数据类型转换21

2.4.1 自动数据类型转换21

2.4.2 强制数据类型转换22

2.5 C++运算符与表达式22

2.5.1 算术运算符与算术表达式22

2.5.2 关系运算符与关系表达式23

2.5.3 逻辑运算符与逻辑表达式24

2.5.4 位运算符与位表达式25

2.5.5 自增（减）操作运算符与 表达式27

2.5.6 赋值与复合运算符29

2.5.7 sizeof 运算符与表达式29

2.5.8 条件运算符与条件表达式30

2.5.9 逗号运算符与逗号表达式30

2.5.10 操作运算符的优先级与结合性 ..31

小结31

综合练习二32

一、选择题32

二、填空题32

三、思考题32

四、上机操作题32

第 3 章 C++程序的结构34

3.1 编译预处理功能34

3.1.1 预处理指令34

3.1.2 文件包含35

3.1.3 宏定义35

3.1.4 带参数的宏定义36

3.1.5 条件编译36

3.2 语句与程序结构概述	38	三、思考题	67
3.2.1 语句	38	四、上机操作题	67
3.2.2 程序的三种基本结构	38	第5章 数组与字符串	68
3.2.3 语句的嵌套深度与缩进风格	40	5.1 数组的概念	68
3.3 数据的输入与输出简介	40	5.2 一维数组	68
3.3.1 输入与输出流	40	5.2.1 一维数组的定义	68
3.3.2 C++输出格式控制	41	5.2.2 一维数组元素的引用	69
小结	42	5.2.3 一维数组的初始化	70
综合练习三	43	5.2.4 一维数组程序举例	70
一、选择题	43	5.3 二维数组	75
二、填空题	43	5.3.1 二维数组的定义	75
三、思考题	43	5.3.2 二维数组元素的引用	76
四、上机操作题	43	5.3.3 二维数组的初始化与赋值	76
第4章 程序流程控制结构	45	5.3.4 二维数组程序举例	77
4.1 顺序结构	45	5.4 字符数组	80
4.2 选择控制结构	45	5.4.1 字符数组的定义和初始化	80
4.2.1 if 语句	46	5.4.2 字符数组的赋值与引用	81
4.2.2 if...else 语句	46	5.4.3 字符数组的输入输出	81
4.2.3 else if 语句	47	5.4.4 字符串处理函数	82
4.2.4 if 语句的嵌套与二义性	47	5.4.5 字符数组应用举例	84
4.2.5 switch 语句	49	小结	85
4.3 循环控制结构	52	综合练习五	86
4.3.1 while 语句	52	一、选择题	86
4.3.2 do...while 语句	54	二、填空题	86
4.3.3 for 语句	55	三、思考题	86
4.3.4 循环的嵌套	57	四、上机操作题	87
4.3.5 三种循环语句的比较	59	第6章 指针与引用	88
4.4 break 语句与 continue 语句	60	6.1 指针的基本概念	88
4.4.1 break 语句	60	6.1.1 内存地址	88
4.4.2 continue 语句	61	6.1.2 指针	89
4.5 goto 语句	62	6.2 指针变量	90
4.6 流程控制应用示例	64	6.2.1 指针变量的定义	90
小结	65	6.2.2 指针变量的使用	90
综合练习四	65	6.3 指针运算	93
一、选择题	65	6.3.1 赋值运算	93
二、填空题	66		

6.3.2 算术运算	94	一、选择题	125
6.3.3 关系运算	95	二、填空题	125
6.4 指针与数组	95	三、思考题	126
6.4.1 指针与一维数组	95	四、上机操作题	126
6.4.2 指针与字符串	100		
6.4.3 指针数组	101		
6.5 动态内存分配 (new 与 delete)	102	第 8 章 函数的高级特性与应用	127
6.5.1 new 运算符	102	8.1 函数与数组	127
6.5.2 delete 运算符	104	8.1.1 数组元素作为函数的参数	127
6.6 引用	105	8.1.2 数组名作为函数的实参	128
6.6.1 引用的声明	105	8.2 函数与指针	129
6.6.2 引用的操作	106	8.2.1 指针作函数的参数	130
6.6.3 引用与指针的区别	106	8.2.2 指针函数	131
6.6.4 其他	107	8.2.3 函数指针	132
小结	107	8.3 内联函数	137
综合练习六	108	8.4 函数的缺省参数	138
一、选择题	108	8.5 函数重载	139
二、填空题	108	8.6 函数的嵌套调用	140
三、思考题	109	8.7 函数的递归调用	141
四、上机操作题	109	小结	142
		综合练习八	143
第 7 章 函数	110	一、选择题	143
7.1 函数定义的一般形式	110	二、填空题	143
7.1.1 函数定义	110	三、思考题	144
7.1.2 函数声明	112	四、上机操作题	144
7.2 函数调用	113		
7.2.1 函数调用方式	115	第 9 章 自定义数据类型	145
7.2.2 参数传递	115	9.1 结构	145
7.3 变量及作用域	118	9.1.1 结构的定义	145
7.3.1 局部变量	119	9.1.2 使用结构	146
7.3.2 全局变量	119	9.1.3 结构与指针	148
7.3.3 静态局部变量	120	9.1.4 结构与数组	149
7.3.4 变量与作用域	121	9.1.5 结构与函数	151
7.3.5 变量的可见性	122	9.2 枚举类型	153
7.4 内部函数和外部函数	123	9.3 用 typedef 声明类型	155
小结	124	小结	156
综合练习七	125	综合练习九	156
		一、选择题	156

二、填空题	157	三、思考题	184
三、思考题	158	四、上机操作题	184
四、上机操作题	158	第 11 章 静态成员与友元	186
第 10 章 类与对象	159	11.1 静态成员	186
10.1 封装、继承、多态	159	11.1.1 静态成员适用的场合	186
10.1.1 封装	159	11.1.2 静态数据成员	187
10.1.2 继承	160	11.1.3 静态成员函数	191
10.1.3 多态	160	11.1.4 静态成员综合举例	193
10.2 类	160	11.2 友元	196
10.2.1 概述	160	11.2.1 引入友元的目的	196
10.2.2 类的定义	161	11.2.2 友元函数	198
10.2.3 类的成员函数	163	11.2.3 友元类	201
10.3 对象	165	小结	205
10.3.1 对象的定义	165	综合练习十一	205
10.3.2 对象成员的访问方法	165	一、选择题	205
10.3.3 再论类的封装	166	二、填空题	206
10.4 this 指针	168	三、思考题	206
10.5 构造函数	168	四、上机操作题	206
10.5.1 构造函数的概念	168	第 12 章 运算符重载	208
10.5.2 缺省构造函数和带参构造 函数	169	12.1 运算符重载的概念	208
10.5.3 重载构造函数	170	12.2 运算符重载的基本方法	209
10.5.4 析构函数	171	12.2.1 运算符函数	209
10.5.5 构造函数与析构函数的执行 顺序	173	12.2.2 重载运算符函数的两种方法	210
10.5.6 拷贝构造函数	174	12.2.3 运算符函数重载的基本形式	210
10.5.7 浅拷贝和深拷贝	175	12.3 运算符重载的一般规则	210
10.6 对象数组	178	12.4 用成员函数重载运算符	211
10.7 对象与指针	180	12.5 用友元函数重载运算符	214
10.8 常对象与常成员	180	12.6 算术运算符、关系运算符的重载	216
10.8.1 常对象	180	12.6.1 构建 Date 类的若干考虑	216
10.8.2 常成员	181	12.6.2 重载+、-及关系运算符	219
小结	183	12.7 重载++、--运算符	223
综合练习十	183	12.8 重载插入运算符<<、提取 运算符>>	225
一、选择题	183	12.9 赋值运算符=的重载	227
二、填空题	184	12.10 下标运算符[]的重载	227

12.11 类型转换.....	229	一、选择题.....	264
12.11.1 转换构造函数.....	229	二、填空题.....	265
12.11.2 转换运算符.....	230	三、思考题.....	265
小结.....	232	四、上机操作题.....	265
综合练习十二.....	232		
一、选择题.....	232		
二、填空题.....	233		
三、思考题.....	233		
四、上机操作题.....	233		
第 13 章 继承与派生.....	235	第 14 章 虚函数与多态性.....	267
13.1 继承与派生的概念.....	235	14.1 静态联编和动态联编的概念.....	267
13.2 派生类的定义.....	238	14.1.1 联编的概念.....	267
13.3 派生类的构成.....	239	14.1.2 C++的联编有两类.....	268
13.3.1 派生类的成员来源.....	239	14.2 虚函数.....	272
13.3.2 派生类的生成过程.....	240	14.2.1 虚函数的声明.....	272
13.4 类的继承方式.....	241	14.2.2 虚函数的功能.....	272
13.4.1 分析派生类成员的访问权限.....	241	14.2.3 虚函数的应用.....	275
13.4.2 一个关于基类私有成员访问 权限的重要结论.....	242	14.2.4 虚函数的使用步骤.....	275
13.4.3 继承方式对访问权限的影响.....	242	14.3 虚析构造函数.....	276
13.5 单一继承下的构造函数与析构 函数.....	250	14.4 纯虚函数和抽象类.....	279
13.5.1 单一继承方式下的构造函数.....	250	14.4.1 纯虚函数.....	279
13.5.2 单一继承方式下的析构造函数.....	251	14.4.2 抽象类.....	280
13.6 多重继承.....	253	14.5 多态性.....	282
13.6.1 多重继承的定义格式.....	253	小结.....	283
13.6.2 多重继承下的构造函数和 析构造函数.....	255	综合练习十四.....	284
13.7 多重继承下的二义性问题.....	257	一、选择题.....	284
13.8 虚基类.....	259	二、填空题.....	285
13.8.1 虚基类的定义格式.....	259	三、思考题.....	285
13.8.2 虚基类及其派生类的构造函数 和析构造函数.....	261	四、上机操作题.....	285
13.9 赋值兼容规则.....	262		
小结.....	263	第 15 章 输入输出流与文件操作.....	286
综合练习十三.....	264	15.1 输入输出流的概念.....	286
		15.2 标准输出流.....	288
		15.2.1 输出流对象.....	288
		15.2.2 格式输出.....	288
		15.2.3 用流成员函数 put()输出字符.....	294
		15.3 标准输入流.....	294
		15.3.1 输入流对象.....	295
		15.3.2 用于字符输入的流成员函数.....	295
		15.3.3 几个与流操作有关的判断 函数.....	296

15.4 文件流与文件操作	296	实验 2 运算符和表达式	314
15.4.1 文件与文件类型	296	实验 3 程序流程控制结构 (一)	317
15.4.2 文件流与文件流对象	297	实验 4 程序流程控制结构 (二)	323
15.4.3 文件操作的流程	298	实验 5 数组的应用	326
15.4.4 文本文件的读写操作	299	实验 6 指针与引用	330
15.4.5 二进制文件的读写操作	303	实验 7 函数	331
15.4.6 文件的随机访问	306	实验 8 函数的高级特性与应用	333
小结	309	实验 9 自定义数据类型	334
综合练习十五	310	实验 10 类与对象	335
一、选择题	310	实验 11 静态成员与友元	337
二、填空题	311	实验 12 运算符重载	341
三、思考题	311	实验 13 继承与派生	344
四、上机操作题	311	实验 14 虚函数与多态性	351
第 16 章 上机实验	312	实验 15 输入输出流与文件操作	354
实验 1 Visual C++ 2005 的使用	312	参考文献	358

第 1 章 C++语言概述

C++是一门优秀的程序设计语言。C++比 C 更容易为人们所学习与掌握，并且以其独特的语言机制在计算机科学领域中得到广泛的应用。

本章学习目标：

- (1) 了解 C++语言的概念、发展历史及 C 与 C++之间的关系。
- (2) 了解 C++语言对程序设计方法的支持，C++编程的特性。
- (3) 了解简单的 C++程序结构。
- (4) 掌握 VC++集成开发环境的使用。

1.1 计算机高级语言发展概况

程序设计语言是表达软件的工具，是人机通信的媒介。从应用的角度，程序设计语言就是一台抽象机器，程序员就是利用这台抽象机器的各种功能编制各种绘声绘色的软件。

本节将让读者了解计算机高级程序设计语言产生的历史背景、现状与发展，以及 C++语言的发展。

1.1.1 计算机高级语言产生的历史背景

计算机语言的发展经历了从低级语言到高级语言的发展过程。低级语言指的是机器语言，这类语言的特点是能直接被计算机理解，机器的处理速度快，但不易为人们所学习和掌握。为了解决这一问题出现了一种介于低级语言与高级语言之间的语言——汇编语言。

汇编语言比机器语言能较好地被人们学习和掌握，但用这种语言编写较大型的程序时效率不高，因此出现了计算机的高级语言，高级语言用人们较熟悉和较好掌握的方法进行程序的编写，使人们编写程序的效率提高，并且程序可维护性也大大地增强。

1.1.2 计算机高级语言的现状与发展

迄今为止，人们已开发了数百种高级语言。但是，其中只有几种获得人们的广泛认可。1954~1957 年，IBM 公司成功开发出 FORTRAN (Formula Translator, 公式翻译器)，用于简化科学与工程应用中涉及的复杂数学计算。目前，FORTRAN 在全球各地的应用仍然非常广泛，尤其是工程计算领域。

COBOL (Common Business Oriented Language, 面向商务的通用语言) 最早由一系列计算机制造商、政府部门及计算机企业级客户于 1959 年联合开发成功。COBOL 主要用于商业应用。商业往往要求精确而高效地处理大量数据。如今，几乎有一半的商业软件仍采用 COBOL 编写。

计算机高级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的发展。高级语言的发展经历了从早期语言到结构化程序设计语言，从面向过程到面向对象的程序设计语言等阶段。

20 世纪 60 年代中后期，软件越来越多，规模越来越大，由于当时软件界主要使用

C 越来越普遍地同多种类型的计算机配合使用，最终衍生了 C 的多种变体。各种变体大致类似，但往往互不兼容。如程序开发者需要编写可移植的程序，以便能在多种平台上运行，这无疑是一处致命的缺陷。1983 年，由美国国家计算机和信息处理标准协会正式成立了技术委员会，目的是“提供一个无歧义的、与机器无关的语言定义”。1989 年，该标准被正式通过。ANSI 与国际标准化组织（ISO）协作，致力于 C 在全世界范围内的标准化普及工作；联合标准文档出版于 1990 年，所以也称为 ANSI/ISO9899:1990。

C++是对 C 的一个扩展，它是 Bjarne Stroustrup 于 20 世纪 80 年代在贝尔实验室开发成功的。C++提供了数量众多的特性，目的是对 C 语言进行改进。但更重要的是，它还引入了崭新的面向对象编程的能力。

C++属于一种混合型语言，即可采用传统的 C 风格进行 C++编程，也可采用新的面向对象的风格，甚至还能在同一个项目中混合采用两种风格。

1.2 面向对象程序设计方法概述

与传统的结构化程序设计语言相比，C++语言更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

20 世纪 60 年代，许多大型软件开发项目都面临严重困境：项目往往不能按时完成，实际成本常常超过预算，最终产品也变得极不稳定。人们逐渐意识到，软件开发的复杂程度远远超过自己的想象。因此，通过当时发起的一系列研究，最终促成了结构化编程概念的问世。这是一种比较严格的程序编写机制，用它可方便地写出思路清晰、高度准确、易于测试/调试以及易于修改的程序。

结构化程序设计方法，在过去的几十年中，一直是程序设计开发的主导思想。具体做法是：为解决某个实际问题确定一种算法，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，其注重的是程序功能的描述。在后面的章节中将讨论结构化编程的原理及一些开发实例。

1.2.2 面向对象的程序设计

随着计算机硬件成本的显著下降，个人计算机变成了一种“普通家电”。但令人遗憾的是，软件开发的成本不降反升，因为程序员们开发出了越来越强大、越来越复杂的应用程序。与此同时，软件开发的基本技术并没有革命性的进步。软件工业正在酝酿一场革命。快速、正确和经济地构建软件，一直是一个难以实现的目标。每次需要新的、更强大的软件时，人们都会以这个目标为准绳。

对象本质是一种可重复使用的软件组件，用于对现实世界中的各种物件进行建模。软件开发者发现，与结构化编程相比，假如采用一种模块化的、面向对象的设计与实现方法，可显著提高软件开发小组的效率。面向对象的程序更易理解、纠正以及修改。

面向对象的程序设计以所处理的数据为中心，而不是以功能为中心来编写程序。数据与功能相比具有较强的稳定性，而面向对象的程序设计与结构化的程序设计最大的区别就在于此，前者的中心是所要处理的数据，而后者的中心是功能。

那么,到底何为对象,它们有何特殊之处呢?实际上,对象技术是一种打包方案,帮助程序员创建有意义的软件单元。许多软件单元都高度集中在特定的应用领域上。于是,人们见到了品种齐全的时间对象、日期对象、支票对象、发票对象、声音对象、视频对象、文件对象、记录对象等等。事实上,世界上的任何一个名词都可以用对象来表示。

人们生活在一个对象的世界里,环顾四周便可体会到这一点。汽车、飞机、人、动物、建筑、红绿灯、电梯等等,一切都是对象。在面向对象的语言问世之前,几乎所有的程序语言(如 FORTRAN、Pascal、Basic 和 C)都将重点放在行动(动词)上,而不是放在物件或对象(名词)上。

程序员生活在一个对象的世界中,但却不得不使用计算机,主要是用动词来进行编程。这种对思维模式的强制转变,加大了编写程序的难度。但现在,由于可选用流行的面向对象语言(如 Java 和 C++以及其他许多语言),程序员可继续生活在一个充斥着形形色色对象的世界,并在使用计算机时,采用自然的、面向对象的方式进行编程。这意味着他们能采用由接触世界而直接感知到的方式编写程序。

1.3 C++语言的语法规则与书写格式

计算机高级语言有其特定的语法规则与书写格式。下面分别介绍 C++语言的相关语法规则与书写格式,并通过一个实例来进行描述。

1.3.1 标识符与关键字

1. 标识符

标识符是用于表示应用程序中的变量、常量、类型、函数、类和对象的名字。在定义变量、类型、函数和类时通过声明一个标识符来创建它,然后就可以在后面的程序中使用该标识符来定义有关的内容。

C++的标识符必须是以下划线或字母开头的,由字母、数字和下划线组成的序列。标识符可以包含任意多个字母、数字或下划线,但只有前 31 个字符有效,并区分大小写字母。注意,在自定义的标识符中最好不要用下划线开头,用下划线开头的使用虽然合法,但由于系统定义的内容多以下划线开头,所以这样做容易引起冲突而产生错误。

下列标识符是不合法的:

```
1st_year #A_security NOT_MOVE!
```

其中数字不能作为标识符和开头字符, #、! 不能用作标识符。

2. 字符集

字符集是组成 C++的基本元素,用 C++语言编写的程序中除了字符型数据外,其他的所有成分只能由字符集中的字符构成。C++语言的字符集由下述三个方面的字符构成:

(1) 英文字符: A~Z, a~z。

(2) 数字字符: 0~9。

(3) 其他字符: !、#、%、~、_、+、=、-、;、:、,、.、/、\、<>、()、[]、{ }、空格。

3. 关键字

关键字是预定义好的具有特殊含义的标识符,关键字的使用只能按照它们预定义好的意义来使用。用户在自定义标识符时,一定不能与这些关键字相同。C++语言中的关键字

如表 1-1 所示。

表 1-1 C++语言中的关键字

关键字					
asm	default	float	operator	static_cast	union
auto	delete	for	private	struct	unsigned
bool	do	friend	protected	switch	using
break	double	goto	public	template	virtual
case	dynamic_cast	if	register	this	void
catch	else	inline	reinterpret_cast	throw	volatile
char	enum	int	return	true	wchar_t
class	explicit	long	short	try	while
const	export	mutable	signed	typedef	const_cast
extern	namespace	sizeof	typeid	continue	false
new	static	typename			

1.3.2 C++语言的应用程序组成

C++语言的应用程序由以下几个基本部分组成：

1. 预处理命令

C++程序经常出现以“#”开头的命令，这些命令称作预处理命令，分为三类：宏定义，文件包含和条件编译。C++语言的“#include <iostream>”就是一个文件包含命令，在编译时系统会将指定的文件与源文件编译成一个可执行文件。

2. 输入与输出

每个程序都要使用输入与输出语句才能同其他的程序或程序员交换信息。C++程序的输入与输出功能是由类函数来实现的，最常用的 C++语言的输入语句形式是“cin>>变量”，而最常用的 C++语言的输出形式是“cout<<变量”。C++语言中有各自专门的输入/输出流库提供各种各样的输入输出功能，以后涉及到相关内容时再详细论述。

3. 语句

语句是 C++程序的基本单位，C++语言的语句有单语句和复合语句两种，单语句由一条命令构成，其最后必须以“;”结尾，即“;”是语句的结束标识。复合语句由多个语句组成，又称语句块，每个复合语句应以左花括号“{”开始，右花括号“}”结尾。

4. 函数

函数是 C++语言程序的基本功能单元，由若干语句组成，若干个函数构成一个文件，若干个文件组成一个 C++应用程序。在 C++若干函数中，有一个函数是最主要的，就是 main() 函数（又称主函数），它标识 C++程序的开始，所以每个 C++程序有且仅有一个 main() 函数。每个函数的函数名可以任意取，原则是“见名知意”，不能与关键字和已有的函数重名。函数体应以“{”开始而以“}”结尾。函数应先定义后调用，定义时函数是各自独立定义，不能嵌套定义。调用时可以嵌套调用。

5. 常量

C++语言允许使用的常量有数字常量（即常数）和符号常量。符号常量的定义有两种：宏定义方式和常量定义方式。如：


```
#define PI 3.1415926
const double PI=3.1415926
```

6. 变量

每个程序设计语言都使用变量，只是方式不同。C++语言的变量种类繁多，使用灵活，是C++语言程序设计优劣的关键所在。

7. 类

类是C++语言中程序的一种基本功能单元，由成员数据和成员函数组成。成员数据描述事物的属性特征，成员函数描述事物的行为特性。类也可以看作是一种由使用者自定义的数据类型。

1.3.3 C++的语法规则与书写格式

C++程序由数据结构和函数构成，函数由语句组成。每个应用程序只能有一个主函数main()作为系统的开始，函数体必须用花括号“{ }”括起来，作为一个整体。

函数体一般由两部分组成：变量定义部分和执行部分。函数体也可以没有任何语句，只有一对花括号“{ }”，称为空函数。

函数不能嵌套定义，只能嵌套调用。即一个函数在定义时，其函数体内不能再定义其他的函数。

一个C++程序总是从main()函数开始执行，与main()函数在程序中的位置无关。

C++语言要求变量必须先定义后调用。

C++程序在书写上可以在一个程序行上写若干条语句，每条语句规定应以“;”结尾。一条语句也可以分成若干行来写，此时要在换行处加上续行标志“\”。

对于不易理解或重要的程序段，可用注释标识符“//”或“/**/”加以注释，“//”表示单行注释，“/* */”表示多行注释。

1.3.4 简单的C++语言程序范例

一个简单的程序：打印一行文字。

C++采用的记号对非程序员来说可能相当陌生。看下面的一个简单的程序，它的功能是打印一行文字。该程序及其屏幕输出如下：

```
1 //print.cpp
2 //C++第一个程序
3 #include <iostream>
4
5 int main()
6 {
7     std::cout<< "Welcome to C++!\n";
8
9     return 0; //指明程序成功结束
10 }
```

输出结果：

```
Welcome to C++!
```

该程序展示了C++语言的一些重要特性。下面将逐行探讨该程序，第1行和第2行：

```
//print.cpp
//C++第一个程序
```

两行均以//开头，这表明//后的内容均为注释。程序员在编写程序时加注释，以便为程序制作文档，并增强程序的可读性。注释还有助于其他人阅读和理解程序员所编写的程序。