NCRE Examination
全国计算机等级考试

教育部考试中心

National Computer Rank Examination

全国计算机等级考试

二级教程

——C++语言程序设计

（2012年版）

高等教育出版社
HIGHER EDUCATION PRESS

全国计算机等级考试二级教程

——C++语言程序设计
（2012年版）

Quanguo Jisuanji Dengji Kaoshi Erji Jiaocheng
——C++ Yuyan Chengxu Sheji

教育部考试中心

高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

由教育部考试中心推出的计算机等级考试是一种客观、公正、科学的专门测试计算机应用人员的计算机知识与技能的全国性考试。它面向社会，服务于社会。

本书根据教育部考试中心最新颁布的“全国计算机等级考试二级C++语言程序设计考试大纲（2007年版修订版）”的要求，在2011年版的基础上修订而成，是在全国计算机等级考试委员会指导下，由教育部考试中心组织编写的计算机等级考试系列教程之一。主要内容包括：C++的数据类型、基本语句、数组指针和引用、函数的使用、类与对象的相关知识，此外还介绍了模板和输入输出流。本书内容精炼，结构合理，便于自学，对读者可能遇到的难点做了十分系统、清楚的阐述，除可以作为计算机等级考试用书外，也可以作为学习C++语言的参考书。

图书在版编目(CIP)数据

全国计算机等级考试二级教程:2012年版.
C++语言程序设计/教育部考试中心编. --北京:高等教育出版社,2011.11(2012.8重印)
ISBN 978-7-04-033909-3

Ⅰ. ①全… Ⅱ. ①教… Ⅲ. ①电子计算机-水平考试-教材②C语言-程序设计-水平考试-教材 Ⅳ. ①TP3

中国版本图书馆CIP数据核字(2011)第238637号

策划编辑 何新权　　责任编辑 何新权　　封面设计 陈 方　　责任校对 胡美萍

出版发行 高等教育出版社
社　　址 北京市西城区德外大街4号
邮政编码 100120
印　　刷 北京明月印务有限责任公司
开　　本 787mm×1092mm 1/16
印　　张 20.5
字　　数 500千字
购书热线 010-58581118
咨询电话 400-810-0598
网　　址 http://www.hep.edu.cn
http://www.hep.com.cn
网上订购 http://www.landraco.com
http://www.landraco.com.cn
版　　次 2011年11月第1版
印　　次 2012年8月第2次印刷
定　　价 38.00元

物 料 号 33909-00

大力推行全国计算机等级考试 为发展知识经济、信息产业和培养计算机专门人才作出贡献

（代　序）

中国科学院院士　北京大学信息与工程科学学部主任
全国计算机等级考试委员会主任委员
杨芙清

当今，人类正在步入一个以智力资源的占有和配置，知识生产、分配和使用为最重要因素的知识经济时代，也就是小平同志提出的“科学技术是第一生产力”的时代。科教是经济发展的基础，知识是人类创新的源泉。基础研究的科学发现、应用研究的原理探索和开发研究的技术发明，三者之间的联系愈来愈紧密，转换周期日趋缩短。世界各国的竞争已成为以经济为基础、以科技（特别是高科技）为先导的综合国力的竞争。

在高科技中，信息科学技术是知识高度密集、学科高度综合，具有科学与技术融合特征的学科。它直接渗透到经济、文化和社会的各个领域，迅速改变着人们的观念、生活和社会的结构，是当代发展知识经济的支柱之一。

在信息科学技术中，微电子是基础，计算机硬件及通信设施是载体，计算机软件是核心。软件是人类知识的固化，是知识经济的基本表征，软件已成为信息时代的新型“物理设施”。人类抽象的经验、知识正逐步由软件予以精确地体现。在信息时代，软件是信息化的核心，国民经济和国防建设、社会发展、人民生活都离不开软件，软件无处不在。软件产业是增长最快的朝阳产业，是具有高额附加值、高投入/高产出、无污染、低能耗的绿色产业。软件产业的发展将推动知识经济的进程，促进从注重量的增长向注重质的提高的方向发展，是典型的知识型产业。软件产业是关系到国家经济安全和文化安全，体现国家综合实力，决定21世纪国际竞争地位的战略性产业。

为了适应知识经济发展的需要，大力推动信息产业的发展，需要在全民中普及计算机的基本知识，广开渠道，培养和造就一批又一批能熟练运用计算机和软件技术的各行各业的专门人才。

1994年，原国家教委（现教育部）推出了全国计算机等级考试，它是一种重视应试人员对计算机和软件的实际掌握能力的考试。它不限制报考人员的学历背景，任何年龄段的人员都可以报考。这就为培养各行各业计算机的应用人才开辟了一条广阔的道路。

1994年是推出计算机等级考试的第一年，当年参加考试的有1万余人；而2010年，年报考人数已近507万人。截至2011年上半年，全国计算机等级考试共开考33次，考生人数累计达4 130万人，其中有1 550万人获得了不同级别的计算机等级证书。

事实说明，鼓励社会各阶层的人士通过各种途径掌握计算机应用技术，并运用等级考试对他们的才干予以认真的、有权威性的认证，是一种人才培养的有效途径，是比较符合我国具体情况

的。等级考试也为用人部门录用和考核人员提供了一种测评手段。从有关公司对等级考试所作的社会抽样调查结果看，不论是管理人员还是应试人员，对该项考试的内容和形式都给予了充分的肯定。

计算机等级考试所取得的良好效果，也同全国各有关单位专家们在等级考试的大纲编写、试题设计、阅卷评分及效果分析等多项工作中所付出的大量心血和辛勤劳动密切相关，他们为这项工作的顺利开展作出了重要的贡献。

计算机与软件技术是一项日新月异的高新技术。计算机等级考试大纲有必要根据计算机与软件技术在近年的新发展，进行适当的修正，从而使等级考试更能反映当前计算机与软件技术的应用实际，使培养计算机应用人才的基础工作更健康地向前发展。

从面临知识经济的机遇与挑战这样一个社会大环境的背景出发，考察全国计算机等级考试，就会看到，这一举措是符合知识经济和信息产业的发展方向的，是值得大力推行的。

我们相信，在21世纪知识经济和信息产业加快发展的形势下，在教育部考试中心的精心组织领导下，在全国各有关专家们的大力配合下，全国计算机等级考试一定会以更新的面貌出现，从而为我国培养计算机应用专门人才的宏大事业作出更多的贡献。

2011年10月

前　言

随着我国计算机应用的进一步普及和深入，人们已经达成了一个共识：计算机知识是当代人类文化的重要组成部分；计算机应用能力是跨世纪人才不可缺少的素质。因此，许多单位把计算机知识和应用能力作为考核、录用工作人员的重要条件；许多人也在努力证实自身在这方面的实力。人们都在寻求一个统一、客观、公正的衡量标准，教育部考试中心组织的“全国计算机等级考试”自1994年举办以来，应试人数逐年递增，是深受社会各界欢迎的计算机考试。

随着计算机应用的发展，等级考试的内容也在不断更新。我们根据教育部考试中心最新颁布的《全国计算机等级考试二级C++语言程序设计考试大纲（2007年版修订版）》的要求，在2011年版的基础上修订而成本教程。本书紧扣考试大纲，内容取舍得当，是一本系统的考试教材。

全书共分十章，内容包括：C++语言概述、C++数据类型、基本语句、数组、指针和引用、函数、对象和类的相关知识、类的继承与派生、多态性、模板和输入输出流。本书的编写力求在体系结构上安排合理、重点突出、难点分散、便于掌握；在语言叙述上注重概念清晰、逻辑性强、便于自学。根据要求，考试分为笔试和上机两部分。为便于读者自我检查，书中各章的最后均配有与笔试题型一致的习题和供上机练习用的编程题。带有“*”号的章节是为了知识的完整性而编写的，考试大纲中不要求，考试中也不涉及。此外，本书在附录中提供了集成开发环境Microsoft Visual C++ 6.0的使用方法，读者可以参照其中的具体步骤进行C++语言编程的上机练习。

本书由教育部考试中心组织编写。第一、二、三章和附录由袁晓洁编写，第五章由晏海华编写，第四章由袁晓洁和晏海华共同编写，第六章由马锐编写，第八、九、十章由李宁编写，第七章由马锐和李宁共同编写。全书由袁晓洁和黄啸波统稿，清华大学郑莉老师对全书进行了全面审阅。在本书的编写和出版过程中，教育部考试中心和高等教育出版社给予了大力支持，在此一并表示衷心感谢。

由于编写时间仓促，难免有疏漏之处，请读者提出宝贵意见，以便修订时改进。

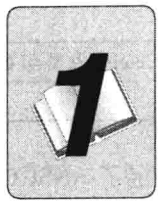
编者

目　录

第 1 章　C++语言概述 …… 1
1.1　C++语言的发展 …… 1
1.2　C++语言的特点 …… 2
1.3　面向对象程序设计 …… 3
1.4　C++语言的基本符号 …… 4
1.5　C++语言的词汇 …… 5
1.5.1　关键字 …… 5
1.5.2　标识符 …… 5
1.5.3　字面常量 …… 6
1.5.4　运算符 …… 6
1.5.5　标点符号 …… 6
1.6　C++程序的基本框架 …… 6
1.7　C++程序的开发过程 …… 8
1.7.1　编辑 …… 8
1.7.2　编译 …… 8
1.7.3　链接 …… 9
1.7.4　运行和调试 …… 9
本章小结 …… 9
习题 1 …… 10
第 2 章　数据类型、运算符和表达式 …… 12
2.1　C++语言的数据类型 …… 12
2.1.1　基本类型 …… 12
2.1.2　基本类型的派生类型 …… 13
2.2　常量 …… 14
2.2.1　逻辑常量 …… 14
2.2.2　字符常量 …… 14
2.2.3　整型常量 …… 16
2.2.4　实型常量 …… 17
2.2.5　枚举常量 …… 18
2.3　变量 …… 18
2.3.1　变量的定义 …… 18
2.3.2　变量的使用方式 …… 20
2.3.3　符号常量声明语句 …… 21
2.4　运算符和表达式 …… 22
2.4.1　运算符和表达式的概念 …… 22
2.4.2　运算类型与运算符 …… 23
2.4.3　赋值运算 …… 23
2.4.4　算术运算符和算术表达式 …… 24
2.4.5　关系运算符和关系表达式 …… 25
2.4.6　逻辑运算符和逻辑表达式 …… 26
2.4.7　位运算 …… 27
2.4.8　其他运算 …… 27
2.4.9　优先级和结合性 …… 30
本章小结 …… 32
习题 2 …… 33
第 3 章　基本控制结构 …… 35
3.1　C++语句 …… 35
3.2　顺序结构 …… 35
3.2.1　声明语句 …… 36
3.2.2　表达式语句 …… 37
3.2.3　基本输入输出 …… 37
3.2.4　复合语句和空语句 …… 41
3.3　选择结构 …… 42
3.3.1　if 语句 …… 42
3.3.2　switch 语句 …… 46
3.4　循环结构 …… 49
3.4.1　for 语句 …… 50
3.4.2　while 语句 …… 52
3.4.3　do…while 语句 …… 54
3.4.4　循环的嵌套 …… 55
3.5　跳转语句 …… 57
3.5.1　break 语句 …… 58
3.5.2　continue 语句 …… 58
3.5.3　return 语句 …… 59
3.5.4　goto 语句 …… 59
本章小结 …… 60
习题 3 …… 61
第 4 章　数组、指针与引用 …… 65
4.1　数组 …… 65
4.1.1　一维数组 …… 65

4.1.2　多维数组 …… 69
4.1.3　字符数组 …… 73
4.2　指针 …… 75
4.2.1　指针和地址 …… 77
4.2.2　指针和数组 …… 81
4.3　引用 …… 84
4.4　动态存储分配 …… 86
本章小结 …… 88
习题4 …… 89
第5章　函数 …… 94
5.1　函数定义 …… 94
5.2　函数调用 …… 96
5.3　函数原型 …… 96
5.4　函数返回类型 …… 98
5.5　函数参数 …… 100
5.5.1　参数的传递方式 …… 100
5.5.2　默认参数 …… 104
5.6　函数重载 …… 105
5.7　内联函数 …… 107
5.8　递归函数 …… 108
5.9　变量的生存周期 …… 111
本章小结 …… 112
习题5 …… 113
第6章　类和对象 …… 118
6.1　类的定义 …… 118
6.1.1　类的定义 …… 118
6.1.2　类成员的访问控制 …… 119
6.1.3　类的数据成员 …… 121
6.1.4　类的成员函数 …… 121
6.2　对象的定义 …… 126
6.2.1　对象的定义 …… 126
6.2.2　对象的成员 …… 127
6.3　构造函数和析构函数 …… 128
6.3.1　构造函数和析构函数的定义 …… 128
6.3.2　默认构造函数和默认析构函数 …… 132
6.3.3　复制构造函数 …… 133
6.4　自由存储对象 …… 136
6.5　this指针 …… 138
6.6　静态成员 …… 139
6.6.1　静态数据成员 …… 139
6.6.2　静态成员函数 …… 141
6.7　常成员 …… 142
6.7.1　常对象 …… 142
6.7.2　常成员函数 …… 143
6.7.3　常数据成员 …… 145
6.8　友元 …… 146
6.8.1　友元函数 …… 146
6.8.2　友元类 …… 149
6.9　对象数组 …… 150
6.10　成员对象 …… 152
本章小结 …… 154
习题6 …… 155
第7章　继承和派生 …… 164
7.1　继承与派生 …… 164
7.1.1　基本概念 …… 164
7.1.2　派生类的定义与构成 …… 166
7.2　派生类对基类成员的访问 …… 166
7.3　派生类的构造函数和析构函数 …… 170
7.3.1　派生类的构造函数 …… 170
7.3.2　派生类的析构函数 …… 173
7.4　多继承与虚基类 …… 175
7.4.1　多继承中的二义性问题 …… 175
7.4.2　虚基类的定义 …… 179
7.4.3　虚基类的构造函数 …… 180
*7.5　子类型关系 …… 182
7.6　虚函数与多态性 …… 186
7.6.1　多态性的概念 …… 186
7.6.2　虚函数 …… 187
*7.6.3　虚析构函数 …… 189
7.6.4　纯虚函数与抽象类 …… 190
本章小结 …… 192
习题7 …… 194
第8章　运算符重载 …… 202
8.1　运算符函数与运算符重载 …… 202
8.2　典型运算符的重载 …… 203
8.2.1　关于分数类 fraction …… 203
8.2.2　重载取负运算符“-” …… 205
8.2.3　重载加法运算符“+” …… 205
8.2.4　重载增量运算符“++” …… 206

8.2.5　重载类型转换符“long” …… 207
8.2.6　重载赋值运算符“=” …… 207
8.2.7　重载复合赋值运算符“+=” …… 210
8.2.8　重载关系运算符“>” …… 210
8.2.9　重载下标访问运算符“[]” …… 210
8.2.10　重载流运算符“>>”和“<<” …… 212
*8.3　运算符重载应注意的几个问题 … 212
本章小结 …… 214
习题8 …… 215
第9章　模板 …… 219
9.1　函数模板 …… 219
9.2　类模板 …… 221
本章小结 …… 223
习题9 …… 223
第10章　C++流 …… 228
10.1　C++流的概念 …… 228
10.1.1　C++流的体系结构 …… 228
10.1.2　预定义流对象 …… 228
10.1.3　提取运算符>>和插入运算符<< …… 228
10.1.4　有格式输入输出和无格式输入输出 …… 230
10.1.5　操作符 …… 230
10.2　输入输出的格式控制 …… 230
10.2.1　默认的输入输出格式 …… 230
10.2.2　格式标志与格式控制 …… 231
10.2.3　输入输出宽度的控制 …… 232
10.2.4　浮点数输出方式的控制 …… 232
10.2.5　输出精度的控制 …… 233
10.2.6　对齐方式的控制 …… 234
10.2.7　小数点处理方式的控制 …… 234
10.2.8　填充字符的控制 …… 235
10.2.9　插入换行符 …… 235
10.2.10　输入输出数制状态的控制 …… 235
10.3　文件流 …… 236
10.3.1　文件流的建立 …… 236
10.3.2　文件流的关闭 …… 237
10.3.3　文件流状态的判别 …… 237
10.3.4　文件流的定位 …… 238
10.3.5　有格式输入输出 …… 238
*10.3.6　无格式输入输出 …… 239
本章小结 …… 242
习题10 …… 242
第11章　上机指导 …… 246
11.1　上机考试系统使用说明 …… 246
11.1.1　上机考试环境 …… 246
11.1.2　上机考试时间 …… 246
11.1.3　上机考试题型及分值 …… 246
11.1.4　上机考试登录 …… 247
11.1.5　试题内容查阅工具的使用 …… 249
11.1.6　考生文件夹和文件的恢复 …… 250
11.2　上机考试内容 …… 251
11.2.1　基本操作题 …… 251
11.2.2　简单应用题 …… 252
11.2.3　综合应用题 …… 255
11.3　上机考试样题 …… 257
附录1　使用Visual C++ 6.0编写标准C++程序 …… 269
附录2　C/C++常用标准库函数 …… 284
附录3　ASCII码表 …… 293
附录4　全国计算机等级考试二级C++语言程序设计考试大纲(2007年版修订版) …… 295
附录5　全国计算机等级考试二级C++语言程序设计样题及参考答案 …… 298
附录6　2011年3月全国计算机等级考试二级笔试试题及参考答案——C++语言程序设计 …… 303
附录7　习题参考答案 …… 312



第1章 C++语言概述

C++是一种优秀的高级程序设计语言，它是以C语言为基础而逐渐发展起来的。C++语言既保留了传统的结构化程序设计方法，又对流行的面向对象程序设计方法提供了完整的支持。此外，C++语言还具有许多C语言不支持的新功能和新特性。

1.1 C++语言的发展

C++起源于C语言。C语言是1972年由美国贝尔实验室的Dennis Ritchie根据B语言开发设计出来的。最初，发明C语言的目的是用它来代替汇编语言为小型机DEC PDP-11编写UNIX操作系统。后来，随着UNIX操作系统的推广，C语言被越来越多的程序设计人员了解和使用。到20世纪70年代末，C语言已经凭借其如下独有的特点风靡了全世界的程序设计领域：

（1）语言简洁、紧凑，使用灵活、方便。

（2）具有丰富的运算符和数据类型。

（3）可以进行许多低级操作，适合开发系统软件。

（4）程序的运行效率高。

（5）代码的可移植性好。

然而，在C语言流行的同时，也暴露出了它的局限性：

（1）类型检查机制相对较弱，这使得程序中的一些错误不能及时被发现。

（2）缺少支持代码重用的语言结构，因此为一个程序所设计的模块，很难再用于其他程序。

（3）不适合开发大型软件，当程序的规模大到一定程度时，维护工作会变得相当复杂。

为了满足开发大规模程序的需要，1980年贝尔实验室的Bjarne Stroustrup和他的同事们开始对C语言进行改进和扩充，把Simula 67（一种早期的面向对象语言）中类的概念引入到C语言，并将改进后的C语言称为“带类的C”（C with class）。1983年夏，“带类的C”被正式命名为“C++”，并于同年7月首次对外发表。1985年由Bjarne Stroustrup编写的《C++程序设计语言》一书出版，这标志着C++ 1.0版本的诞生。此后，贝尔实验室于1989年和1993年分别推出了C++ 2.0版本和C++ 3.0版本。表1.1列出了各版本中C++语言所添加的一些新特性。

表1.1 C++语言支持的新特性

版　本	在C语言基础上添加的新特性
带类的C	类和派生类，公有成员和私有成员，构造函数和析构函数，友元，内联函数，赋值运算符的重载

续表

版　本	在C语言基础上添加的新特性
C++ 1.0	虚函数,函数运算符的重载,引用,常量
C++ 2.0	类的保护成员,多重继承,赋值和初始化的递归定义,抽象类,静态成员函数,const成员函数
C++ 3.0	模板,异常,类的嵌套,名字空间

经过对C++语言的三次修订后,美国国家标准委员会(ANSI)于1994年制定了ANSI C++标准草案,这个草案最终于1998年被国际标准化组织(ISO)批准为国际标准(ISO/IEC 14882)。C++就是这样在不断地发展和完善中走过了二十多年的历史。至今,它仍然是一种充满活力的程序设计语言。

1.2 C++语言的特点

在众多的高级程序设计语言中,C++能够取得成功的原因在于它有着许多与众不同的特点。

1. C++是一种面向对象的程序设计语言

C++语言支持几乎所有的面向对象程序设计特征。可以说,C++语言集中体现了近20年来在程序设计和软件开发领域出现的新思想和新技术,这主要包括:

(1) 抽象数据类型。

(2) 封装和信息隐藏。

(3) 以继承和派生方式实现程序的重用。

(4) 以运算符重载和虚函数来实现多态性。

(5) 以模板来实现类型的参数化。

2. C++是程序员和软件开发者在实践中创造的

一般的高级程序设计语言是由计算机科学家在科研和教学环境中设计出来的,然而,C++语言却是由从事实际系统开发工作的程序员在实践中创造的。因此,C++往往从编写实际程序的角度出发,为程序员提供了各种实用、灵活、高效的语言特性。正是这些良好的特性使得C++在很多领域,特别是大规模系统程序的开发方面,得到了广泛的应用。目前,许多成功的大型软件都是使用C++编写的。

3. C++是C语言的超集

所谓"C++是C语言的超集"是指C++中包含C语言的全部语法特征。因此,每一个用C语言编写的程序都是一个C++程序。C++语言的设计宗旨就是在不改变C语言语法规则的基础上扩充新的特性。C++与C语言的关系可以用图1.1表示。

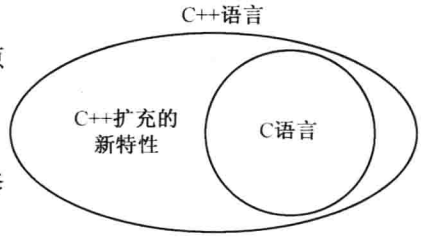


图1.1 C++与C语言的关系

实际上,能够很好地兼容C语言正是C++取得成功的原因之一,这是因为:

(1) C++继承了C语言简明、高效、灵活等众多优点。

(2) 以前使用C语言编写的大批软件可以不加任何修改,直接在C++开发环境下维护。

(3) C语言程序员只需要学习C++扩充的新特性,就可以

很快地使用C++编写程序。

由于具有上述特点，C++已经开始取代 C 语言，被广泛地应用于各种领域的程序设计工作中。实践表明，对于中型和大型程序的开发工作，使用C++的效果要比 C 语言好得多。C++正在从软件的可靠性、可重用性、可扩充性、可维护性等方面体现出它的优越性。

1.3 面向对象程序设计

面向对象程序设计（Object-Oriented Programming，简称 OOP）是 20 世纪 80 年代发展起来的一种程序设计方法。它通过模拟现实世界中的事物和关系，利用抽象、分类、归纳等方法来构造软件系统。

在面向对象程序设计出现之前，人们一直采用结构化程序设计（Structured Programming，简称 SP）来解决实际问题。结构化程序设计是面向过程的，其主要思想是将功能分解并逐步求精。Pascal 语言和 C 语言都很好地体现了结构化程序设计的思想。

按照结构化程序设计的要求，当需要解决一个复杂的问题时，首先应将它按功能划分为若干个小问题，每个小问题又可以按功能划分为若干个更小的问题，依此类推，直到最低一层的问题较容易用程序实现为止；然后将所有的小问题全部解决并把它们组合起来，复杂的问题就迎刃而解了。然而到了 20 世纪 80 年代末，随着所要开发程序规模的增大，结构化程序设计的一些缺点就显得越来越突出，这主要表现为：

（1）数据和算法的一致性差。在结构化程序设计中，数据与处理数据的算法是相互分离的。当数据量增大时，程序会变得越来越难理解。如果根据需要而改变某一项数据时，处理此数据的所有算法都要作相应的修改，这就很容易使算法与数据出现不一致的现象。这样的程序是很难修改和维护的。

（2）程序的可重用性差。在电子技术中，要实现某种功能往往有标准的元器件供选择，而不需要自己去设计发明。这就体现出了可重用的思想，即某种通用功能由事先设计好的标准部件来实现。如果在程序设计中可重用性高，那么在很大程度上可以节省人力和物力的浪费。但是结构化程序设计并不支持可重用性，这就使得程序员在开发软件时每次都从零做起，重复着许多同样的工作。

而现实世界中的实际情况恰恰与结构化程序设计思想不同。现实世界中每一种事物都具有一些属性来描述自身的特征，同时具有一些操作来改变自身状态。例如，一辆汽车可以用有型号、颜色、载重量、行驶速度等信息进行描述，这些都是这辆汽车的属性；而开动汽车使它前进、后退、左转、右转等，都是对汽车状态的操作。这样，全部属性和操作的集合就定义了这种汽车的类型。显然，在程序设计中属性对应于数据，操作对应于算法。因此，将数据和处理数据的算法捆绑成一个整体就定义了一种事物的类型。事物类型是一种抽象的概念，只有属于该类型的一个事物实例才是具体可见的，这个实例就叫做“对象”，而事物类型被称为“类”，它们都是面向对象程序设计的基础。

C++是一种面向对象的程序设计语言，它充分支持面向对象思想中的三个主要特征：

1. 封装性

封装性是指将数据和算法捆绑成一个整体，这个整体就是对象，描述对象的数据被封装在其

内部。如果需要存取数据,可以通过对象提供的算法来进行操作,而无需知道对象内部的数据是如何表示和存储的。这种思想被称为信息隐藏。例如,使用者不必知道一台电视机内部电路的具体构造和工作原理,就可以用它来收看电视节目。封装性和数据隐藏从根本上解决了结构化程序设计中数据和算法一致性差的问题。

C++语言通过建立用户定义类型——“类”,来支持封装性和信息隐藏。用户定义的类一旦建立,就可看成是一个完全封装的实体,可以作为一个整体单元来使用。类的内部数据表示被隐藏起来,类的用户不需要知道类内数据的表示方法,只需执行类对外提供的算法,就可以完成某项功能。

2. 继承性

继承性是指一种事物保留了另一种事物的全部特征,并且具有自身的独有特征。例如,建筑工程师已经设计出了一座普通楼房的图纸,后来又需要设计办公楼和居民楼。这时,可以有两种选择:一是从零开始,分别重新设计办公楼和居民楼;二是在普通楼房图纸的基础上分别添加新的功能,使它成为办公楼和居民楼。工程师当然不想总是从头做起,因为办公楼和居民楼都属于楼房,它们都具有楼房的全部特征。既然已经成功地设计出普通楼房的图纸,就不必再费力劳神地重复设计普通楼房了。实际上,工程师在设计具有新功能的楼房时,重复地使用着普通楼房的概念。这种思想被称为可重用。

C++语言采用继承来支持重用,程序可以在现有类型的基础上扩展功能来定义新类型。新类型是从现有类型中派生出来的,因此被称为派生类。

3. 多态性

多态性是指当多种事物继承自一种事物时,同一种操作在它们之间表现出不同的行为。例如,在一个使用面向对象思想编写的绘图程序中可能含有四种类型的对象,它们分别用于表示抽象概念——形状和具体概念——三角形、矩形、圆形。其中三角形、矩形、圆形对象都继承了形状对象的全部特征,并且三者都有一个名为“显示”的操作。但当用户对这三种不同的具体形状分别执行“显示”操作时,会在屏幕上得到三种不同的图案。这个例子就说明了多态性。

C++语言中使用函数重载、模板、虚函数等概念来支持多态性。

1.4 C++语言的基本符号

组成语言的最小元素是基本符号。汉语中有4 000多个常用单字,英语中有26个拉丁字母,它们都是组成各自语言的基本符号。同样,作为一种计算机程序设计语言,C++也是由基本符号组成的。

C++语言中的基本符号可以分为3类:

1. 字母

包括大写英文字母A~Z和小写英文字母a~z共52个符号。

2. 数字

包括0~9共10个符号。

3. 特殊符号

包括:+ - * / = , . _ : ; ? \ " ' ~ | ! # % & ( ) [ ] { } ^ < >和“空格”共30个符号。

这三类符号共计92个,它们组成了C++语言的基本符号集合。

1.5 C++语言的词汇

基本符号本身一般没有什么含义，而由它们按照一定规则组合成的单词却能表达出某种语义。使用C++语言编写的程序正是由符合规则的单词组成的。那么，什么样的单词才算符合规则呢？答案是：只有下面列出的五类单词才是C++语言中的合法词汇。

1.5.1 关键字

关键字也称为保留字，它是由C++语言本身预先定义好的一类单词。表 1.2 和表 1.3 列出了C++的关键字。其中表 1.2 是 ANSI C 标准规定的 32 个关键字，表 1.3 是 ANSI C++标准补充的 29 个关键字。

表 1.2 ANSI C 标准规定的关键字

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

表 1.3 ANSI C++标准补充的关键字

bool	catch	class	const _ cast	delete	dynamic _ cast
explicit	false	friend	inline	mutable	namespace
new	operator	private	protected	public	reinterpret _ case
static _ cast	template	this	throw	true	try
typeid	typename	using	virtual	wchar _ t	

每个关键字在C++语言中都具有特殊的含义，并实现着一定的功能。所以，不能将上述关键字再当做其他类型的单词使用。

1.5.2 标识符

标识符是用户为程序中各种需要命名的“元素”所起的名字。这些“元素”包括：变量、符号常量、函数、函数的参数、结构、类、对象等。标识符的组成要符合一定的规则：

（1）标识符是一个以字母或下划线开头的，由字母、数字、下划线组成的字符串。例如，Hello，var2，myBook，func _ 1，_ TEST _ H 都是合法的标识符；而 012，3var，A * B，$ sys，pay. due 都是非法的标识符。

（2）标识符不能与任意一个关键字同名。例如，if、case、for 等都不能作为标识符使用。

（3）标识符中的字母区分大小写。例如，Abc 和 abc 是两个不同的标识符。

（4）标识符不宜过长。C++语言不限制标识符的长度，但多数编译器只识别前 32 个字符，如果程序中的标识符超过了这个长度，超出的部分被忽略不计。

1.5.3 字面常量

常量是指在程序中固定不变的值。常量一般有两种表示形式,即符号常量和字面常量。例如,在“pi 等于 3.1416”这句话中,pi 是一个符号常量,pi 的值 3.1416 称为字面常量。C++语言中符号常量的名字就是一个标识符,而字面常量却是一类特殊的单词。字面常量分为整型、浮点型、字符型和字符串型四类。它们的具体命名规则将在第 2 章中介绍。

1.5.4 运算符

运算符是对程序中的数据进行操作的一类单词。C++语言中运算符的种类非常丰富,其中有:

(1) 单字符组成的运算符,例如:+,-,*,/等。

(2) 双字符组成的运算符,例如:++,<=,&&,*=,->等。

(3) 三个字符组成的运算符,例如:<<=,>>=等。

(4) 关键字运算符:new,delete,sizeof。

各种运算符的具体使用方法将在第 2 章中介绍。

1.5.5 标点符号

标点符号是在程序中起分割内容和界定范围作用的一类单词,表 1.4 列出了C++语言中的标点符号。

表 1.4 C++语言的标点符号

标点符号	描述	标点符号	描述
(空格)	语句中各成分之间的分割符	{ (左花括号)	复合语句的开始标记符
; (分号)	语句的结束符	} (右花括号)	复合语句的结束标记符
' (单引号)	字符常量的起止标记符	// (双斜杠)	行注释的开始标记符
" (双引号)	字符串常量的起止标记符	/* (斜杠和星号)	块注释的开始标记符
# (井字号)	预处理命令的开始标记符	*/ (星号和斜杠)	块注释的结束标记符

1.6 C++程序的基本框架

对C++语言词汇的介绍,可以从微观上认识C++程序的基本组成单元;而要对C++程序有一个宏观上的总体把握,就要进一步了解程序的基本框架。为便于理解,先来看一个最简单的C++程序。

例 一个最简单的C++程序

```
// ex1_1.cpp
#include <iostream>
int main( )
```

```
{
    std::cout << "This is a simple C++ program. \n";
    return 0;
}
```

此程序的运行结果是在屏幕上显示：

```
This is a simple C++ program.
```

上述程序虽然只有7行，但它却包含了每一个C++程序都要具备的几个基本组成部分。由于这是本书中第一个完整的C++程序，这里对它进行逐行解释：

第一行：**// ex1 _ 1. cpp**

这是一个注释行。注释是程序编写者为读者作的一种说明，有助于提高程序的可读性。C++中提供了“行”和“块”两种注释方法。行注释的内容从双斜杠“//”开始到本行末尾结束，例如，“// ex1 _ 1. cpp”就是行注释；块注释的内容从符号“/*”开始到符号“*/”结束，例如：

```
/* only one line */
```

和

```
/* this is the first line
   this is the second line
   this is the third line */
```

都是块注释。本行注释中的内容为源程序文件的名称——“ex1 _ 1. cpp”，通常C++源程序文件以 . cpp 为扩展名。

第二行：**#include <iostream>**

这是一条预处理命令。C++中每一个以符号“#”开头的行都是预处理命令。使用预处理命令可以更好地进行程序的组织。本行预处理命令“#include <iostream>”的作用是将头文件 iostream 的内容加入到程序中。iostream 是一个C++标准头文件，其中定义了一些输入输出流对象。

第三行：**int main ( )**

本行是主函数的声明。主函数是所有C++程序开始执行的入口。无论主函数处于程序中的什么位置，其中的代码总是最先被执行。按照C++语言的规定，每个程序都必须有且仅有一个主函数，主函数的名称必须为 main。

main 前面的 int 表示主函数 main 将返回一个 int 类型的值。int 代表整型值，它是C++中的一个基本数据类型。

跟在 main 后面的圆括号“( )”说明它是一个函数。在C++中所有的函数名称后面都紧跟着一对圆括号，其中可以没有内容（即没有函数的参数，例如本行），也可以包含函数的参数。

第四、七行：在主函数 main 的声明之后用花括号“{}”括起来的是函数主体部分。

第五行：**std::cout << "This is a simple C++ program. \n";**

本行是一条C++语句，它完成了此程序的主要功能，即向屏幕上输出一行字符串。cout 是C++中的标准输出流对象，它通常代表计算机的屏幕。cout 在标准头文件 iostream 中被声明（注：标识符 cout 位于 std 名字空间中，须用前缀 std::进行修饰），因此要使用它就必须先包含此文件（见程序的第2行）。“<<”是输出操作符，功能是将它右边的内容输出到它左边的指定设备

上。这里,要输出的内容是用双引号括起来的字符串“This is a simple C++ program. \n”,其中的字符“\n”是换行符。本行末尾的分号“;”表示这条语句的结束。在C++中所有的语句都要以分号结束。

第六行:**return 0;**

本行是 return 语句,它的功能是使主函数 main 结束并将整数 0 返回给运行此程序的操作系统。返回整数 0 表示程序在执行过程中没有发生任何错误而正常结束。

实际上,按照C++标准,本行是可以省略的。如果在主函数 main 中不写“return 0;”语句,程序正常结束时也会自动向操作系统返回 0。

1.7 C++程序的开发过程

开发一个C++程序的过程通常包括编辑、编译、链接、运行和调试等步骤。目前有许多软件产品可以帮助我们完成C++程序的开发。例如,在 Windows 平台下有 Microsoft 公司的 Visual C++和 Borland 公司的C++ Builder;在 Linux 平台下有 GUN 的 gcc 和 gdb 等。本节将以 Microsoft 公司的 Visual C++ 6.0(简称 VC6)为工具来介绍C++程序的开发过程。关于使用 VC6 集成开发环境编写标准C++程序的具体方法请参见本书附录 A。

1.7.1 编辑

编辑是C++程序开发过程的第一步,它主要包括程序文本的输入和修改。任何一种文本编辑器都可以完成这项工作。

在 VC6 集成开发环境中,用户可以使用编辑窗口来进行C++程序的编辑工作。VC6 的编辑窗口是专门为编辑C++程序而设计的,它提供了包括语法亮色、调用提示、自动缩进、查找和替换等在内的一系列功能,使用起来十分方便。

当用户完成了C++程序的编辑时,应将输入的程序文本保存为以 .cpp 为扩展名的文件(保存C++头文件时应以 .h 为扩展名)。

1.7.2 编译

C++是一种高级程序设计语言,它的语法规则与汇编语言和机器语言相比更接近人类自然语言(如英语)的习惯。然而,计算机能够“看”懂的唯一语言是机器指令。因此,当我们要让计算机“看”懂一个C++程序时,就必须使用一种叫做“编译器”的工具,将这个C++程序“翻译”成机器指令。

编译器所做的工作实际上是一种由高级语言到机器指令的等价变换。用户提供给编译器的输入信息称为源程序代码,它是用某种高级语言编写的程序文本。编译器对源程序代码进行一系列处理后最终产生的输出结果称为目标代码,它是某种计算机的机器指令,并且在功能上与源程序代码完全等价。保存源程序代码和目标代码的文件分别称为源程序文件和目标文件。由源程序文件到目标文件的转换过程就称为编译。

实际上,在进行编译之前还要完成一项称为“预处理”的工作。它的目的是根据程序中的预处理命令对源程序代码做出相应的处理,或为编译器提供一些提示信息。程序中的注释在