

白艳梅 李宏光 段惠东 主 编
步春红 梁永先 崔丽文 副主编

C++语言程序设计

计算机科学与技术专业实践系列教材

清华大学出版社



计算机科学与技术专业实践系列

C++语言程序设计

白艳梅 李宏光 段惠东 主 编
步春红 梁永先 崔丽文 副主编

清华大学出版社

内 容 简 介

本书以面向工程实践和编程能力训练为主,系统讲述 C++ 语言的基础知识、基本语法及编程方法,具有较强的应用性。

全书共 10 章,主要内容包括: C++ 概述, C++ 的数据类型、运算符和表达式,语句和预处理,函数,数组和字符串,指针和引用,类和对象,继承和派生,多态性和虚函数,以及流类库和输入输出操作。

全书中内容安排由浅入深,讲解方法通俗易懂,每章设置“小结”,对关键知识点和要求掌握的知识点进行归纳说明,并辅以课后习题强化关键知识点,使读者能够正确、深入地理解问题。程序调试和运行环境为 Windows 平台下的 Visual C++ 6.0。另外,为帮助读者理解教材内容,强化实践动手能力,本书还配有《C++ 程序设计题解与实验指导》。

本书可作为高等学校 C++ 语言程序设计课程的教学用书,也可作为计算机基础程序设计的培训教材,同时还可作为 C++ 语言的自学指导书和参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C++ 语言程序设计/白艳梅等主编. —北京: 清华大学出版社, 2015

计算机科学与技术专业实践系列教材

ISBN 978-7-302-40091-2

I. ①C… II. ①白… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 089649 号

责任编辑: 白立军 战晓雷

封面设计: 傅瑞学

责任校对: 白 蕾

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 三河市吉祥印务有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 25

字 数: 606 千字

版 次: 2015 年 8 月第 1 版

印 次: 2015 年 8 月第 1 次印刷

印 数: 1~2000

定 价: 44.50 元

序

在当今世界,以计算机技术、通信技术和控制技术为代表的 3C 技术正迅猛发展,以 Internet 为代表的全球信息基础设施的建设标志着人类社会已进入信息时代。应用型人才培养是社会发展和高等教育发展的必然要求,经济社会发展迫切需要高等学校培养出知识、能力、素质等诸方面都适应社会需要的不同层次的应用型人才,满足信息化社会建设的需求。实施高等教育名校建设工程,对大力发展高等教育,指导高等教育特色发展,全面提高教育质量,增强高等教育的竞争力和服务经济社会能力具有重大意义。

罗伯特·加涅是美国教育心理学家,加涅将认知学习理论应用于教学过程,加涅理论中的引出作业和提供反馈是一种教学策略。从应用型人才培养的角度来说,学生实践能力的提升是一个重要问题,需要学校和教师采取一些有效手段来增强学生的实践能力,树立以学生为本的观念,尊重学生的个性特点,因材施教,拓展学生对于课程、专业的选择空间。

我们在应用型人才培养教学内容和课程体系改革方面做了多年的研究和实践。我们提出的课程内容设置方案,是以推进人文与自然的融合,满足学生能力、兴趣、个性、人格全面发展的需要,强化学生的实践能力和创新能力培养为目标,实施教学方式、教学内容、考核机制的全面改革,在培养学生的信息能力和信息素养方面具有先导作用,为计算机课程教学内容和课程体系改革设计了一个全新的框架。为了反映我们多年来进行应用型人才培养教学内容和课程体系改革的综合成果,我们组织教师编写了以下课程的教材: J2EE 企业级开发(Struts+Spring+Hibernate 整合技术)、ASP.NET 网站设计、C++ 语言程序设计及实验、基于工作任务的 Java 程序设计及实验。这些教材以应用型、技能型人才培养为目标,以重点专业建设为平台,围绕教育教学改革、创新人才培养、提高人才培养质量的教育发展理念展开。每部教材都是由应用型名校计算机专业教师或者计算机实验教学示范中心专业教师编著完成的。

在这套教材的编写过程中,我们得到许多专家的精心指点和热情帮助。教育部高等学校计算机类专业教学指导委员会先后三次在鲁东大学召开计算机基础课程教学研讨会,有清华大学、北京大学、中国人民大学、复旦大学、浙江大学、南京大学、中国科学技术大学等近百所高校参加,专家学者向我们提出了很多宝贵意见。

这套教材的出版得到清华大学出版社的大力支持,正是他们精益求精的工作,才使这一系统工程得以顺利完成,在此表示衷心感谢。

刘启明
于鲁东大学
2015 年 5 月

前 言

本书是我们多年来进行应用型人才培养教学内容和课程体系改革的综合成果。本书内容以面向工程实践和编程能力训练为主,具有较强的可读性和应用性,为计算机程序设计课程教学内容和课程体系改革构建了一个全新的框架。

C++语言是在C语言基础上扩充了面向对象机制而形成的一种面向对象程序设计语言,它继承了C语言的全部优点和功能,同时还支持面向对象程序设计。C++语言现在已经成为介绍面向对象程序设计的首选语言,被普遍地应用于科学技术和日常生活的各个领域。

本书共分10章。第1章为C++语言概述,介绍C++语言的发展和特点以及Visual C++ 6.0编译系统的使用方法等;第2章为C++语言程序设计基础,介绍基本数据类型、运算符及表达式、数据类型转换等;第3章主要包括算法、选择结构语句和选择结构程序设计、循环语句和循环结构程序设计以及预处理等;第4章主要包括函数概念和定义、函数间的参数传递、函数调用、变量的存储类别等;第5章主要包括一维数组、二维数组介绍,字符数组、常用字符处理函数的基本使用方法等;第6章主要包括指针的定义、指针的操作、指针与数组、指针与函数、引用及其应用等;第7章和第8章详细叙述了C++语言支持面向对象程序设计的基本方法,包括类、对象、派生类、继承;第9章介绍多态性和虚函数;第10章介绍流类库和输入输出操作。

本书所讲述的内容既有对C语言重要内容的复习,又有C++语言的主要内容,并且以讲解面向对象的三大特性(封装性、继承性和多态性)为重点,系统地介绍了C++语言编程所用的全部内容。

本书内容安排由浅入深,讲解方法通俗易懂。本书以丰富的例题讲解操作方法、验证语法规则,例题针对性强,读者通过学会一个例题,可以掌握一个概念、一种操作方法或一种编程技巧。

本书的每一章都设置了“小结”,对关键知识点和要求掌握的知识点进行了归纳说明,并辅以大量的课后习题强化关键知识点,使读者能够正确、深入地理解问题。

从应用型人才培养的角度来说,学生的实践能力提升是一个重要问题,需要学校和教师采取有效的教学手段,树立以学生为本的观念,尊重学生的个性特点,因材施教。本书对此做了有益的探索,内容从基础原理、基本开发技术和应用3个层面来组织,使学生对计算机程序设计概念由感性的、功能性的认识上升到内涵的、机理性的认识。

另外,为帮助读者理解教材内容,强化实践动手能力,本书还配有《C++程序设计题解与实验指导》。

本书中的所有程序都在Windows平台的Visual C++ 6.0版本的编译系统中调试

通过。

本书由白艳梅、李宏光、段惠东主编，步春红、梁永先、崔丽文副主编。

在本书的编写过程中，我们查阅了大量有关 C++ 语言程序设计的文献资料，在此对文献资料的作者表示感谢。尽管我们为本书编写付出了很大努力，并希望本书能成为一部精品，但限于作者水平，加之时间仓促，书中难免有疏漏和不妥之处，敬请广大读者不吝赐教。

编 者

2015 年 5 月

目 录

第 1 章 C++ 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和发展	1
1.1.2 面向对象的基本概念	1
1.2 C++ 的发展历史	2
1.3 C++ 的特点	2
1.4 C++ 语言与 C 语言	3
1.4.1 C++ 语言对 C 语言的继承	3
1.4.2 C++ 语言对 C 语言的改进	3
1.5 C++ 程序结构与书写格式	4
1.5.1 简单的 C++ 语言程序	4
1.5.2 C++ 程序的构成	8
1.5.3 C++ 程序的书写格式	9
1.6 C++ 程序的实现	9
1.6.1 简单 C++ 程序的编写和运行过程	9
1.6.2 C++ 单文件程序调试的一般过程及调试手段	10
1.6.3 C++ 多文件程序的实现	15
本章小结	19
习题	19
第 2 章 数据类型、运算符和表达式	21
2.1 标识符	21
2.2 数据类型	21
2.2.1 整型	22
2.2.2 实型	25
2.2.3 字符型	25
2.3 常量和变量	25
2.3.1 整型常量和整型变量	27
2.3.2 实型常量和实型变量	28
2.3.3 字符常量和字符变量	29
2.3.4 字符串常量	29
2.4 常用函数	30
2.5 算术运算符和算术表达式	31

2.5.1	算术运算符	32
2.5.2	算术表达式	33
2.6	关系运算符和关系表达式	33
2.6.1	关系运算符	33
2.6.2	关系表达式	34
2.7	逻辑运算符和逻辑表达式	35
2.7.1	逻辑运算符	35
2.7.2	逻辑表达式	35
2.8	赋值运算符和赋值表达式	37
2.8.1	赋值运算符	37
2.8.2	赋值表达式	38
2.9	逗号运算符和逗号表达式	38
2.10	数据类型转换	39
2.10.1	数据类型的隐式转换	39
2.10.2	数据类型的显式转换	40
2.11	运算符的优先级和结合性	40
2.11.1	运算符的优先级	40
2.11.2	运算符的结合性	42
2.12	数据的输入、输出	42
2.12.1	数据的输出函数	42
2.12.2	数据的输入函数	46
2.12.3	输入流 cin	49
2.12.4	输出流 cout	51
	本章小结	58
	习题	58
第 3 章	语句和预处理	61
3.1	算法	61
3.1.1	算法及其表示	61
3.1.2	算法的设计过程	63
3.2	if 语句	64
3.2.1	if 语句的基本形式	64
3.2.2	if 语句的嵌套	69
3.3	条件运算符和条件运算表达式	70
3.4	switch 语句	72
3.5	选择程序结构	74
3.6	循环语句	78
3.6.1	循环程序的组成	79
3.6.2	循环程序的分类	79
3.7	while 语句和 do-while 语句	79

3.7.1	用法	79
3.7.2	执行过程	80
3.7.3	循环的嵌套	82
3.7.4	应用举例	82
3.8	for 语句	84
3.8.1	用法	84
3.8.2	执行过程	86
3.8.3	循环的嵌套	86
3.8.4	for 语句的变化形式	88
3.8.5	应用举例	91
3.9	循环的控制	97
3.9.1	复杂循环的控制条件	97
3.9.2	break 语句和 continue 语句	97
3.9.3	goto 语句	100
3.9.4	应用举例	101
3.10	预处理	104
3.10.1	文件包含命令	104
3.10.2	宏定义命令	105
3.10.3	条件编译命令	108
	本章小结	110
	习题	110
第 4 章	函数	125
4.1	函数的定义与调用	125
4.1.1	函数的定义	125
4.1.2	函数的调用	129
4.2	函数间参数的传递	132
4.3	变量的属性	134
4.3.1	局部变量和全局变量	134
4.3.2	动态存储方式和静态存储方式	136
4.3.3	变量的存储类别	137
4.4	函数的嵌套调用与递归调用	141
4.4.1	函数的嵌套调用	141
4.4.2	函数的递归调用	143
	本章小结	147
	习题	147
第 5 章	数组和字符串	152
5.1	一维数组	152
5.1.1	一维数组的定义	152

5.1.2	一维数组的初始化	153
5.1.3	一维数组元素的引用	154
5.1.4	一维数组的应用	156
5.2	二维数组	161
5.2.1	二维数组的定义	161
5.2.2	二维数组的初始化	162
5.2.3	二维数组元素的引用	163
5.2.4	二维数组的应用	163
5.3	数组作为函数参数	164
5.3.1	一维数组与函数	164
5.3.2	二维数组与函数	166
5.4	字符数组	168
5.4.1	字符数组与字符串	168
5.4.2	字符串的输入与输出	169
5.4.3	字符串处理函数	175
	本章小结	178
	习题	178
第 6 章	指针和引用	182
6.1	指针的概念	182
6.1.1	指针和指针变量	182
6.1.2	直接访问数据和间接访问数据	182
6.2	指针的操作	183
6.2.1	指针变量的定义	183
6.2.2	指针变量的赋值	184
6.2.3	指针运算符	185
6.3	指针与数组	188
6.3.1	一维数组与指针	188
6.3.2	二维数组和指针	191
6.3.3	字符串和指针	193
6.4	指针与函数	197
6.4.1	指针参数与函数的地址传送调用	197
6.4.2	返回指针的函数	201
6.4.3	指向函数的指针变量	202
6.4.4	使用参数的 main 函数	203
6.5	引用	205
6.5.1	引用的概念	205
6.5.2	引用的应用	206
6.5.3	引用与指针的区别	210
	本章小结	210

习题	212
第7章 类和对象	215
7.1 面向对象程序设计概述	215
7.1.1 面向过程和面向对象	215
7.1.2 面向对象程序设计的基本特点	215
7.1.3 类和对象	216
7.2 类的定义	216
7.2.1 类的声明格式	216
7.2.2 定义类的注意事项	219
7.2.3 类与结构体	219
7.3 成员函数	221
7.3.1 成员函数概述	221
7.3.2 成员函数的定义	221
7.4 对象	225
7.4.1 类与对象的关系	225
7.4.2 对象的基本概念	225
7.4.3 对象的定义	225
7.4.4 对象成员的表示方法	227
7.4.5 对象的赋值	230
7.4.6 简单应用举例	231
7.5 构造函数	235
7.5.1 数据成员的初始化	235
7.5.2 构造函数的定义	236
7.5.3 构造函数的特点	237
7.5.4 构造函数的分类	238
7.5.5 用参数初始化表对数据成员初始化	250
7.5.6 带有默认参数的构造函数	251
7.5.7 构造函数的重载	255
7.6 析构函数	257
7.6.1 析构函数的概念	257
7.6.2 默认的析构函数	259
7.7 this 指针	264
7.8 共用数据的保护	267
7.8.1 常对象	267
7.8.2 常对象成员	270
7.8.3 常对象指针	276
7.8.4 常对象引用	282
7.9 静态成员	283
7.9.1 静态数据成员	283

7.9.2 静态成员函数	289
7.10 友元	291
7.10.1 友元函数	292
7.10.2 友元成员	295
7.10.3 友元类	296
本章小结	297
习题	298
第8章 继承和派生	305
8.1 继承和派生的概念	305
8.2 单继承派生类的定义	306
8.2.1 派生类的声明	306
8.2.2 派生类的构成	307
8.3 派生类的继承方式	308
8.3.1 类的公有继承	309
8.3.2 类的私有继承	311
8.3.3 类的保护继承	313
8.4 派生类的构造函数和析构函数	314
8.4.1 派生类的构造函数	314
8.4.2 有子对象的派生类的构造函数	315
8.4.3 派生类的析构函数	316
8.5 多重继承	318
8.5.1 多重继承的声明	318
8.5.2 多重继承的构造函数和析构函数	319
8.5.3 多重继承的二义性	320
8.6 虚基类	324
本章小结	325
习题	325
第9章 多态性和虚函数	338
9.1 多态性	338
9.1.1 多态的概念	338
9.1.2 多态性的一个典型例子	338
9.2 基类和派生类之间的转换	342
9.3 虚函数	345
9.4 纯虚函数与抽象类	347
本章小结	350
习题	350
第10章 流类库和输入输出操作	355
10.1 输入输出流的概念	355

10.2	标准输入输出流·····	356
10.2.1	标准输入流的使用·····	357
10.2.2	标准输出流的使用·····	360
10.3	文件的输入和输出·····	365
10.3.1	文件打开和关闭·····	365
10.3.2	文本文件的读写·····	366
10.3.3	二进制文件的读写·····	367
10.3.4	文件的随机存取·····	368
	本章小结·····	370
	习题·····	370
附录 A	字符的 ASCII 表·····	373
附录 B	Microsoft Visual C++ 6.0 集成开发工具简介 ·····	375
B.1	Visual C++ 6.0 概述·····	375
B.2	Visual C++ 6.0 的启动及其主窗口简介·····	375
B.3	菜单功能介绍·····	375
B.4	调试菜单功能介绍·····	381
B.5	上下文关联菜单·····	382
B.6	工具栏·····	383
B.7	工作区窗口·····	383
B.8	输出窗口·····	383
B.9	编辑窗口·····	383
	参考文献·····	384

第 1 章 C++ 概述

C++ 语言是一种广泛使用的计算机语言,现在已经成为介绍面向对象程序设计的首选语言。它是在 C 语言的基础之上扩充了面向对象机制而形成的一种面向对象程序设计语言,它继承了 C 语言的全部优点和功能,同时还支持面向对象程序设计。学会 C++ 语言对学习其他面向对象的程序设计语言很有帮助。

1.1 面向对象的概念

1.1.1 面向对象方法的由来和发展

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。

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作各自不同的对象,再把具有相同特性的一些对象归属为一个类。

面向对象方法是计算机科学发展的要求。这种方法满足了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的客观需要。

1.1.2 面向对象的基本概念

面向对象编程语言简称为 OOP。由于 OOP 的对象、类具有高度的抽象性,所以能很好地表达任何复杂的数据类型,也允许程序员灵活地定义自己所需要的数据类型。类本身具有很完整的封装性,可以使用它作为编程中的模块单元,满足模块独立自主的需求。再加上继承性和多态性,更有助于简化大型软件和大量重复定义的模块,增强了模块的可重用性,提高了软件的可靠性,缩短了软件的开发周期。

1. 对象

对象是现实世界中客观存在的某种事物。对象是一种相对独立的实体,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。

对象是计算机内存中的一块区域,通过将内存分块,每个模块(即对象)在功能上相互之间保持相对独立。当对象的一个成员函数被调用时,对象执行其内部的代码来响应这个调用,这使对象呈现出一定的行为。行为及其结果就是该对象的功能。

2. 类

类是人们对于客观事物的高度抽象。

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

述包括属性和行为两大部分。

3. 面向对象

面向对象是一种认识世界的方法,也是一种程序设计方法。面向对象的观点认为,客观世界是由各种各样的实体,也就是对象组成的。每种对象都有自己的内部状态和运动规律,不同对象间的相互联系和相互作用就构成了各种不同的系统,并进而构成整个客观世界。按照这样的思想设计程序,就是面向对象的程序设计。“面向对象”不仅作为一种技术,更作为一种方法贯穿于软件设计的各个阶段。

4. 面向对象程序设计

面向对象程序设计是在面向过程的程序设计基础上的质的飞跃。面向对象方法的产生是计算机科学发展的要求。面向对象的技术在系统程序设计、数据库及多媒体应用等领域都得到广泛应用。

1.2 C++ 的发展历史

自从 1946 年第一台电子数字计算机 ENIAC 问世以来,计算机应用领域的不断扩大促进了计算机技术的高速发展,尤其近年来计算机的硬件和软件都日新月异。作为应用计算机的一种工具——程序设计语言,也得到不断的充实和完善。每年都有新的程序设计语言问世,老的程序设计语言不断地更新换代。

C++ 是在 C 语言的基础上发展和完善的。1980 年贝尔实验室的 Bjarne Stroustrup 博士及其同事开始对 C 语言进行改进和扩充,最初称为“带类的 C”,1983 年取名为 C++,以后又经过不断完善和发展成为目前的 C++。C++ 包含了 C 语言的全部特征、属性和优点,同时,C++ 添加了对面向对象程序设计(Object Oriented Programming, OOP)的完全支持。C 语言是建立 C++ 的基础,这就使许多 C 代码稍加修改甚至不经修改就可以为 C++ 所用,用 C 语言编写的众多的库函数和实用软件可以用于 C++ 中。另外,用 C++ 编写的程序可读性更好,代码结构更为合理,可以更直接地在程序中映射问题空间的结构。更重要的是,C 程序员仅需学习 C++ 的新特征就可以很快地用 C++ 编写程序。

1.3 C++ 的特点

C++ 是在 C 语言的基础上,弥补了 C 语言存在的一些缺陷,增加了面向对象的特征,所形成的一种过程性与面向对象性相结合的程序设计语言。

C++ 具有以下特点。

1. 抽象性

面向对象鼓励程序员以抽象的观点看待程序,即程序是由一组对象组成的。可以将一组对象的共同特征进一步抽象出来,从而形成“类”的概念。

2. 封装性

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。封装体具有独立性和隐藏性。一个封装体与外部联系只能通过有限的接口。

封装性就是指一组数据和与这组数据有关的操作集合组装在一起,形成一个能动的实

体,也就是对象。数据封装就是给数据提供了与外界联系的标准接口,无论是谁,只有通过这些接口,使用规范的方式,才能访问这些数据。

3. 继承性

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。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。

继承性是指从已有的对象类型出发建立一种新的对象类型,使它继承原对象的特点和功能。这种思想是面向对象设计方法的主要贡献。

4. 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。

不同的对象可以调用相同名称的函数,并可导致完全不同的行为的现象称为多态性。

1.4 C++ 语言与 C 语言

1.4.1 C++ 语言对 C 语言的继承

C++ 语言由 C 语言发展而来,因此与 C 语言兼容,C 语言中的基本数据类型、运算符与表达式、基本程序结构(顺序、选择与循环)、数组、函数、指针、结构体、联合体、枚举、文件操作等,C++ 完全兼容。C 语言是 C++ 语言的一个子集。

C++ 语言克服了 C 语言本身存在的缺点,并保持了 C 语言简洁、高效,与汇编语言接近的特点,同时还保留了某些 C 语言的面向过程的特性,具有 C 语言的简练明了的风格。实际上,C++ 语言是一种不完全的面向对象的程序设计语言。

1.4.2 C++ 语言对 C 语言的改进

1980 年,贝尔实验室的 Bjarne Stroustrup 博士及其同事对 C 语言进行了改进和扩充,并把 Simula 67 中类的概念引入到 C 语言中。并在 1983 年由 Rick Maseitti 提议正式命名为 C++ (C Plus Plus)。后来,又把运算符的重载、引用、虚函数等功能加入到 C++ 中,使 C++ 的功能日趋完善。

C++ 对 C 语言的改进主要表现在以下几个方面:

- (1) C++ 语言中规定函数定义时必须指出类型。
- (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。
- (3) C++ 语言规定凡是从高类型向低类型转换时都需加强制转换。
- (4) C++ 语言中符号常量建议使用 const 关键字来定义。
- (5) C++ 语言中引进了内联函数。
- (6) C++ 语言允许设置函数参数的默认值。
- (7) C++ 语言引进了函数重载和运算符重载。
- (8) C++ 语言引进了引用概念,使用引用作函数的参数和返回值。
- (9) C++ 语言提供了与 C 语言不同的 I/O 流类库,方便了输入输出操作。
- (10) C++ 语言为方便操作还采取了其他措施。

总的来看,C++ 语言对 C 语言的“增强”表现在以下两个方面:

(1) 对于面向过程的机制,C++ 对 C 语言的功能做了扩充。

(2) C++ 增加了面向对象的机制。

面向对象程序设计是针对开发较大规模的程序而提出来的,目的是提高软件开发的效率。不要把面向对象和面向过程对立起来,两者不是矛盾的,而是各有用途、互为补充的。

学习 C++ ,既要会利用 C++ 进行面向过程的结构化程序设计,也要会利用 C++ 进行面向对象的程序设计。本书既介绍 C++ 在面向过程程序设计中的应用,也介绍 C++ 在面向对象程序设计中的应用。

需要强调的是,早期的 C++ 版本保留了很多 C 语言的传统用法,本教材严格遵守 ANSI C++ 标准编写程序,遵从 C++ 的编程风格。

1.5 C++ 程序结构与书写格式

本节通过几个较简单的 C++ 语言程序,了解 C++ 语言的结构及书写格式。这里给出的程序是不包含面向对象特征的 C++ 程序。

1.5.1 简单的 C++ 语言程序

为了进一步了解 C++ 语言的基本特点,下面介绍几个较简单的 C++ 语言程序。

例 1.1 在屏幕上输出以下一行信息:

```
I am a student.
```

编写程序:

```
/*
在屏幕上显示下列内容
I am a student.
*/
#include<iostream> //以上为程序功能注释或说明部分
using namespace std; //头文件,不带后缀,用 cout 输出时需要用此文件
int main() //使用命名空间 std
{ //定义主函数,返回值为 int 型
    cout<<"I am a student.\n"; //函数体开始标志
    return 0; //函数体部分,输出流,在屏幕上打印引号内的字符串
} //函数执行完毕返回函数值 0。分号是一条完整语句的结束符
//函数结束标志
```

程序运行结果:

```
I am a student.
Press any key to continue
```

以上是在 Visual C++ 6.0 环境下运行程序时屏幕上得到的运行结果。其中第一行是程序运行后输出的结果,第二行“Press any key to continue”是 Visual C++ 6.0 系统在输出运行结果后自动输出的一行信息,告诉用户:“如果想继续进行下一步,请按任意键。”当用户按任意键后,屏幕上不再显示运行结果,而返回程序窗口,以便进行下一步工作。