新世纪应用型高等教育
软件专业系列规划教材

C++ 程序设计

新世纪应用型高等教育教材编审委员会　组编
主 编　李秉璋　罗　烨

大连理工大学出版社

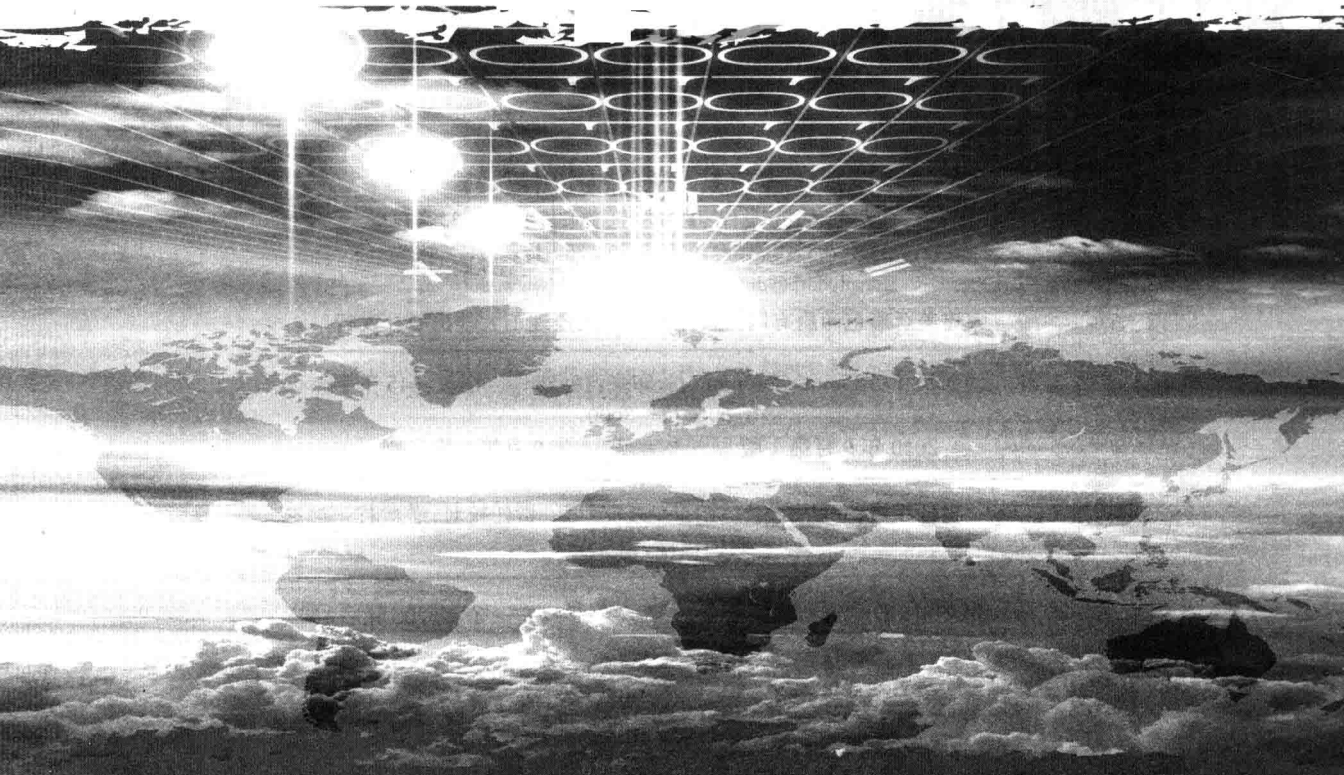
新世纪应用型高等教育
软件专业系列规划教材

C++ 程序设计

新世纪应用型高等教育教材编审委员会 组编

主 编 李秉璋 罗 烨

副主编 李延军



大连理工大学出版社

图书在版编目(CIP)数据

C++程序设计 / 李秉璋，罗烨主编. — 大连 ：大连理工大学出版社，2010.9
新世纪应用型高等教育软件专业系列规划教材
ISBN 978-7-5611-5813-5

Ⅰ. ①C… Ⅱ. ①李… ②罗… 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(2010)第191417号

大连理工大学出版社出版
地址：大连市软件园路80号　邮政编码：116023
发行：0411-84708842　邮购：0411-84703636　传真：0411-84701466
E-mail：dutp@dutp.cn　URL：http://www.dutp.cn
大连美跃彩色印刷有限公司印刷　　大连理工大学出版社发行

幅面尺寸：185mm×260mm　印张：19.25　字数：445千字
印数：1～3000
2010年9月第1版　2010年9月第1次印刷

责任编辑：潘弘喆　　责任校对：潘素君
封面设计：张　莹

ISBN 978-7-5611-5813-5　　定　价：36.00元

前　言

在面向对象程序设计语言中，C++语言是最流行的语言之一。C++从C语言继承发展而来，因此语法严谨、数据类型丰富、运行效率高。同时C++既支持结构化的程序设计方法，也支持面向对象的程序设计方法。因此，C++语言已经成为各高等学校理工类专业的首选计算机语言。

作为“程序设计基础”、“面向对象程序设计”课程的教学研究、改革内容，本教材综合了CC2001、中国计算机科学与技术学科教程和计算机学科专业规范中关于程序设计、算法等相关知识单元的要求，根据应用型本科人才对程序设计能力的要求，结合多年讲授程序设计语言、面向对象技术等课程的教学经验编写而成。

本教材以C++语言为载体，结合C++语言的新技术、新发展，在讲授与C兼容的面向过程的内容后，重点介绍了面向对象的重要概念、技术，包括类与对象、继承与派生、虚函数与多态性、模板、异常处理等。与本教材配套的《C++程序设计实验与实训指导》则从提高学生的面向对象程序设计能力出发，安排了题型丰富的课后练习、测验、课程实验以及综合性的实训项目。

本教材编写宗旨是：面向应用，重在实践，通过课程学习，切实提高学生使用面向对象技术解决实际问题的能力。为了体现这一宗旨，全书的内容体系安排特点是：教学内容循序渐进，所有概念、技术均有例题分析讲解。从第2章开始，每章最后设有单独一节，安排一个综合本章主要知识点，内容上前后衔接的综合性案例。这样到本书最后一章，案例就成为一个融C++主要概念、技术，功能较为完整的程序系统。

本教材的内容可以分成两大部分：第1至第4章为第一部分，主要对C++程序设计思想、面向过程程序设计的基本内容进行介绍，其中包括：C++语言成分、数据类型、表达式、流程控制、函数、文件结构等；第5至第10章为第二部分，重点介绍面向对象程序设计的基本内容，包括类与对象、继承、多态、模板、输入/输出流、异常处理等。

使用本教材，可以根据不同专业对学生面向对象程序设计能力的不同要求，安排在一个学年两个学期进行，其中第一学

期学习第1至第4章，第二学期学习第5至第10章。对于已经学习过面向过程程序设计的学生(如学习过C语言)，也可安排在一个学期完成，重点学习第5至第10章的内容。教材中加“*”的内容对于初学者可以略去。

本教材的教学安排建议如下：

第一部分56学时(包括理论课、习题课、实验课)，其中：第1章10学时，第2章12学时，第3章16学时，第4章18学时。

第二部分56学时(包括理论课、习题课、实验课)，其中：第5章16学时，第6章6学时，第7章12学时，第8章10学时，第9章8学时，第10章4学时。

本教材由长期从事程序设计教学的教师在多年授课的基础上，吸收众多C++教材的特色，一些学校C++课程教学改革经验，结合精品课程建设要求编写而成。本教材由李秉璋、罗烨任主编，李延军任副主编，具体分工为：第1至第4章由罗烨编写，第5至第9章由李秉璋编写，第10章及附录由李延军编写；全书由李秉璋审阅并统稿。

在本教材编写过程中，得到了众多专家学者、同行、同事的指导帮助。南京大学陈家骏教授以渊博的专业理论和丰富的教学经验，对应用型人才C++教材内容选取、章节安排提出了建设性意见；江苏技术师范学院潘瑜、吴访升、叶飞跃、白凤娥等教授则从应用型人才培养规格、人才能力结构方面提出了许多意见和建议。值此教材出版之际，一并向他们表示衷心感谢。

尽管我们在本教材的编写方面做了很多努力，但由于作者水平所限，不当之处在所难免，恳请各位读者批评指正，并将意见和建议及时反馈给我们，以便下次修订时改进。

所有意见和建议请发往：gzjckfb@163.com
欢迎访问我们的网站：http://www.dutpgz.cn
联系电话：0411-84707492　84706104

编　者
2010年9月

目　录

第1章

C++基础

C++是广泛使用的支持多种程序设计方法的语言。本章首先介绍C++语言的发展历史及其特点，结合实例介绍C++程序的基本构成；然后介绍C++中的基本词法单位、数据类型、相关运算，以及常量、变量、表达式、语句等基础知识；最后介绍简单的输入/输出方法。

学习目标

1. 了解C++语言的发展历史及其特点；
2. 掌握C++语言程序的构成和开发过程；
3. 掌握C++语言的基本词法单位；
4. 理解数据类型的概念，掌握变量及常量的概念、定义与使用方法；
5. 掌握常用运算符的含义、优先级、结合性与使用方法；
6. 掌握表达式的构成规则和使用；
7. 理解类型转换概念，掌握数据类型转换规则；
8. 掌握基本输入/输出方法。

1.1 C++概述

1.1.1 程序设计语言

人类的社会生活中，“自然语言”是人与人之间用来交流的工具，是由语音、词汇和语法构成的系统。而“程序设计语言”是人指挥计算机工作的工具，是计算机可以识别的语言，用于描述解决问题的方法，供计算机阅读和执行。

计算机的工作是依靠程序来控制的，程序是为实现特定目标或解决特定问题用计算机语言编写的命令序列的集合，程序规定了计算机执行的动作及其顺序。

在计算机诞生的初期，程序员使用机器语言编写程序，机器语言由计算机硬件系统能够识别的二进制指令组成。我们将机器语言称为低级语言。但是对于人类来说，低级语言晦涩难懂，难以记忆，软件开发效率低、难度大。于是人类发明了汇编语言，它将机器指令映射为一些可以被人读懂的助记符，如ADD、SUB等。用汇编语言编写的程序需要经过翻译才能转换成计算机硬件系统可以识别的机器指令，这种翻译工具称为编译器。尽管如此，汇编语言仍然是低级语言，和人的思维相差甚远，必须详尽地描述计算机

的任何操作，抽象层次低，程序员需要考虑大量的人机交互细节。

高级语言的出现是计算机编程语言的一大进步，它使得计算机程序设计语言不再过度依赖于某种特定的计算机或环境，这是因为高级语言在不同的平台上会被编译成不同的计算机语言，而不是直接被计算机执行。高级语言忽略了计算机的硬件区别，屏蔽了机器的细节，提高了语言的抽象层次，更接近人类的语言，因而易学、易用、易维护。用高级语言编写的程序称为“源程序”，和汇编语言的程序一样，计算机不能直接识别源程序，必须将程序编译成二进制的机器指令才能在计算机上运行。

常见的高级语言有：Fortran，Basic，Pascal，C，C++，C#，Java 等，不同的语言有不同的应用范围。

1.1.2 程序设计

程序设计是根据特定的问题，使用某种程序设计语言，设计出计算机能够执行的指令序列。程序设计是一项创造性的工作，根据任务主要需完成如下两方面工作：

(1)数据描述：数据描述是把被处理的信息描述成计算机可以接收的数据形式，如整数、实数、字符、数组等。

信息可以用人工或自动化装置进行记录、解释和处理。使用计算机进行信息处理时，这些信息必须转换成可以被计算机识别的“数据”，如数字、文字、图形、声音、动画等。不管什么数据，计算机都以二进制形式进行存储和加工处理。数据是信息的载体，信息依靠数据来表达。

(2)数据处理：数据处理是指对数据进行输入、输出、整理、计算、存储、维护等一系列的操作。数据处理的目的是为了提取所需的数据成分，以获得有用的资料。

一些程序员，尤其是程序设计初学者，常常认为程序设计就是用某种程序设计语言编写代码，这其实是错误的认识。上述工作应该被看成是编码，它是在程序设计完成之后才开始的。程序设计需要用一定的方法来指导，对问题如何进行抽象和分解，对程序如何进行组织，才能使程序的可维护性、可读性、稳定性、效率等更好，是程序设计方法研究的问题。计算机对问题的求解方式通常采用数学模型抽象的方法。随着社会科学的发展，人们需要计算机处理的问题越来越复杂，计算机工作者不断寻求简捷可靠的软件开发方法。从结构化程序设计到面向对象程序设计，体现了程序设计理论、方法的不断发展。

学习C++语言，不仅可以掌握一种实用的计算机软件设计工具，更重要的是通过该课程的学习，掌握结构化程序设计和面向对象程序设计的基本方法，为进一步的学习和应用打下良好基础。

1.1.3 C++语言的发展历史

C++语言是在 C 语言的基础上发展起来的。C 语言是在 20 世纪 70 年代初由美国 AT&T 贝尔实验室的 Dennis Richie 等人在 B 语言的基础上通过增加数据类型设计出的一种语言。最初 C 语言作为 UNIX 操作系统的开发语言被人们所认识。20 世纪 70 年代末，随着微型计算机的发展，C 语言开始移植到非 UNIX 环境中，并逐步脱离 UNIX 系

统成为一种独立的程序设计语言。1988 年，美国国家标准协会(American National Standards Institute，ANSI)对 C 语言进行了标准化，产生了 ANSI C，成为以后众多 C 语言版本的基础。

C 语言与当时的其他高级语言相比，具有可以直接访问物理地址的优点，提供类似于汇编语言的系统资源操纵能力及程序执行效率，适合编写系统软件，而又比汇编语言可读性强、使用方便，所以得到了广泛的应用。

但是从程序设计方法的角度看，C 语言同当时常用的其他高级语言一样，都是面向过程的，以数据和数据的处理过程为设计核心。这种设计方法随着问题复杂性的增加和程序规模的扩大逐步显露出局限性。为了适应大规模程序设计的需要，从 20 世纪 70 年代开始，程序设计的焦点就由结构化程序设计方法转移到了面向对象程序设计。

C++就是由 C 发展而来的以面向对象为主要特征的语言。它是 20 世纪 80 年代初由贝尔实验室的 Bjarne Stroustrup 博士发明的，最初称为“带类的 C”，1983 年正式命名为C++。1989 年，负责 C++标准化的委员会 ANSI X3J16 挂牌成立，后简称为 J16。1991 年，负责 C++ 语言国际标准化的技术委员会工作组 ISO/IEC JTC1/SC22/WG21 召开了第一次会议，开始进行 C++ 国际标准化的工作。从此，ANSI 和国际标准化组织(International Standardization Organization，ISO) 的标准化工作保持同步，互相协调。1998 年，通过了C++的国际标准，即 ISO/IEC 14882-1998。2003 年进行了小幅修订，版本号为 ISO/IEC 14882-2003。

遗憾的是，由于C++语言过于复杂，经历了多年的演变之后，仍几乎没有完全符合标准的编译器。可喜的是，各种编译器也在努力提高对C++标准的兼容性。目前比较流行的C++集成开发环境有 Microsoft Visual C++，Borland C++，C++ Builder 等。本教材采用的是 Visual C++ 6.0。

1.1.4 C++语言的特点

C++从 C 语言发展而来，比 C 更好，其特点主要包括：

(1)作为 C 语言的超集，C++继承了 C 的所有优点，与 C 语言兼容，支持结构化的程序设计。熟悉 C 语言的程序员，能够迅速地掌握C++语言。

(2)C++对 C 的数据类型做了扩充，修补了 C 语言中的一些漏洞，提供更好的类型检查和编译时的分析。

(3)生成目标程序质量高，程序执行效率高。一般来说，用面向对象的C++语言编写的程序执行速度与 C 语言程序不相上下。

(4)支持面向对象的程序设计，通过类和对象的概念把数据和对数据的操作封装在一起，模块的独立性更强。通过派生、多态以及模板机制来实现软件的复用。

(5)提供了异常处理机制，简化了程序的出错处理。C++中，出错处理程序不必与正常的代码紧密结合，从而提高了程序的可靠性和可读性。

C++提高了程序的可读性、可靠性、可重用性和可维护性，更适合大型复杂软件的开发。

C++与 C 完全兼容，很多用 C 编写的库函数和应用程序都可以为C++所用，这使得

C++和面向对象技术很快得到推广。但正是由于与C兼容，因此C++不是纯面向对象语言，它既支持面向对象程序设计，又支持面向过程程序设计。在大型复杂程序的设计过程中，应当注意用面向对象的思想进行设计，以更好地发挥C++的优势。

1.2 简单的C++程序实例

C++的程序结构由注释、编译预处理指令和程序主体组成。下面通过一个简单的程序来分析C++程序的基本构成及主要特点。

【例 1-1】 输入圆的半径，编程计算圆周长、圆面积、球体积并将其显示出来。

```
//ex1_1.cpp 求圆周长、圆面积、球体积
#include <iostream>                          //包含头文件
using namespace std;                         //使用名字空间
int main(){
    int radius;                              //定义变量
    double perimeter,area;
    const double PI=3.14;                    //定义常变量
    cout<<"半径= ";
    cin>>radius;                             //输入半径
    perimeter=2 * PI * radius;               //计算圆周长
    area=PI * radius * radius;               //计算圆面积
    cout<<"圆周长= "<<perimeter<<endl;       //输出圆周长
    cout<<"圆面积= "<<area<<endl;
    cout<<"球体积= "<<4/3.0 * PI * radius * radius * radius<<endl;
    return 0;
}
```

该源程序文件名为 ex1_1.cpp，经编译、链接生成可执行程序，运行后，显示器首先显示：

半径=

用户从键盘上输入一个整数 3 并按 Enter 键，显示器上将显示结果：

圆周长= 18.84

圆面积= 28.26

球体积=113.04

下面分析该程序的结构。

1. 注释

注释是为了改善程序的可读性，在编译、运行时不起作用，可以放在程序的任何位置。C++注释有两种形式：一种是在“//”之后的内容，一直到本行结束；另一种是“/*”和“*/”之间的内容，注释内容可占多行。

2. 编译预处理指令

“#”号后为编译预处理指令，本例中的“include”称为文件包含指令，指出程序要使

用外部文件 iostream。它指示编译器在对程序进行预处理时，将文件 iostream 中的代码嵌入到该指令所在处，使其成为本程序文件的一个组成部分。iostream 是系统定义的头文件，其中定义了和输入输出操作有关的内容，如 cin、cout。在C++程序中如果使用了系统中提供的一些功能，就必须包含相关的头文件。编译预处理是C++提供的组织程序的工具，有关内容将在第 3 章中介绍。“using namespace std;”表示这些被包含文件放在称为“std”的名字空间域中，有关名字空间域的概念将在第 5 章中介绍。

C++编译系统提供的头文件有两类，一类是标准C++库的头文件，这些头文件没有扩展名；另一类是 C 语言风格的头文件，文件的扩展名是“. h”，例如 iostream. h，使用该风格的头文件不需要加“using namespace std;”，本书的例题使用的都是标准C++库的头文件。

3. 程序主体

程序主体通常由一些数据的说明以及若干函数所组成，任何C++的程序都是由一个或多个函数组成的。本例由一个称为主函数的“main()”函数组成。在组成程序的若干个函数中，主函数有且只能有一个，它是程序执行的入口，也是程序运行的结束。

每一个函数都包含了函数头和函数体，本例中的“int main()”为函数头，而由“{”和“}”括起来的部分称为函数体。main 函数头中 int 的作用是声明函数的返回值类型为整型，此时必须在函数中至少包含一条 return 语句(本例为函数体内最后一条语句)，该语句的作用是函数执行结束时，向操作系统返回一个零值。

对函数的描述由函数体即“{ }”中的语句序列完成，每个语句以“;”结束。一个语句可能是定义或声明一个变量，如“int radius;”，也可能是得到一个数值的计算步骤，如“perimeter=2 * PI * radius;”。cin 表示标准输入流对象，>>是提取运算符，用于将用户从键盘中输入的值保存到其后面的变量中，cout 表示标准输出流对象，<<是插入运算符，用于将其后面的内容输出到显示器上。

C++严格区分大小写。语法上虽然不严格限制程序的书写格式，但从提高可读性的角度出发，程序书写应采用左缩进格式，呈锯齿状。一般一行只写一条语句。

1.3 C++程序开发过程

C++程序开发通常要经过 5 个阶段，包括：编辑、编译预处理、编译、连接、运行与调试。

1. 编辑

编辑阶段的任务是编辑源程序，源程序是使用C++语言规范书写的程序。C++源程序文件通常带有“. h”、“. cpp”扩展名。其中“. cpp”是标准的C++源程序扩展名，“. h”是头文件扩展名。一个C++程序可以有多个源程序文件。对C++源程序的编辑可以使用多种编辑器，如文本编辑器或C++集成开发环境。

2. 编译预处理

在编译器开始翻译源程序之前，预处理器会自动执行源程序中的预处理语句(命令)。这些预处理语句是规定在编译之前执行的语句，其处理包括：将其他源程序文件包

括到要编译的文件中，执行各种文字替换等。

3. 编译

由高级语言编写的C++源程序无法被计算机直接识别和执行，必须先转换成二进制形式的文件。由源程序转换成二进制代码的过程称为编译，这个过程由编译器来完成。编译过程分为词法分析、语法分析、代码生成这 3 个步骤。在进行词法和语法分析过程中如果发现错误，编译结束后会提示出错信息，必须修改源程序，纠正错误后才能继续下面的工作。当编译结束时没有出现任何错误，就会生成目标程序(或称目标代码)。目标程序可以是机器指令代码，也可用汇编语言或其他中间语言表示。目标程序文件的扩展名为“.obj”。

4. 连接

虽然目标程序是由可执行的机器指令组成的，但并不能由计算机直接执行。因为C++程序的文件中通常包含了对系统定义函数和数据的引用，也可能包含了本程序其他文件中自定义的函数和数据的引用。一个源程序文件编译生成目标代码时，这些地方通常是“空缺”的，连接器的功能就是将多个源程序文件生成的目标文件代码和系统库文件的代码连接起来，将“空缺”补上，生成可执行代码，并存储可执行代码，即 Windows 系统下的可执行文件，其扩展名为“.exe”。

现在一些C++系统产品，如 Microsoft Visual C++与 Borland C++，将程序的编辑、编译和连接集成在一个集成环境中。在这个环境中，编译与连接可以一起进行，但编译与连接是两个不同的阶段，当连接出错时，C++系统会显示连接错误。程序连接通过后，生成可执行文件。运行时，可执行文件由操作系统装入内存，然后 CPU 从内存中取出程序执行。

5. 调试

在程序开发过程的各个阶段都可能出现错误，编译阶段出现的错误称为编译错误；连接阶段出现的错误称为连接错误；在程序运行过程出现的错误可能是逻辑错误或运行错误。逻辑错误和运行错误可通过C++系统提供的调试工具 debug 帮助发现，然后修改源程序，改正错误。目前C++系统都提供源代码级的调试工具，可直接对源程序进行调试。

在C++系统下开发程序的过程如图 1-1 所示。

1.4 C++的词法单位

所有语言系统都是由字符集和规则组成的。“字符”是语言不可细分的最基本语法单位。按照规则，由字符可以组成“词”，由词组成“表达式”、“语句”，又由各种语句构成“函数”、“程序”等。本节介绍C++的字符集，以及各种词法单位。

1. C++字符集

ASCII(American Standard Code for Information Interchange)码字符集是计算机领域中常用的西文字符集，它包括英文字母、阿拉伯数字等在内的 128 个字符。每个 ASCII 码字符的存储占用一个字节。附录 A 给出了 ASCII 字符表。