高职高专计算机系列规划教材

电子·教育

C++程序设计

（第3版）

周志德　侯正昌　主编　　刘德强　主审

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

高职高专计算机系列规划教材

C++程序设计

（第3版）

周志德　侯正昌　主编

刘德强　主审

電子工業出版社

Publishing House of Electronics Industry

北京・BEIJING

内 容 简 介

本书共 13 章，前两章为 C++概述及数据类型和表达式介绍。第 3、4 章叙述了三种基本程序结构、流程控制语句和数组。第 5、6 章讨论了函数的定义和调用、函数的嵌套调用和递归调用、变量的存储类型、内联函数、函数的重载、编译预处理中的宏定义、“文件包含”处理与条件编译。第 7 章讲解了指针变量、指针数组、指向一维数组的指针变量、返回指针值的函数、函数指针变量、new 和 delete 运算符、引用。第 8 章介绍枚举型、结构体与链表。第 9、10 章讲述了类和对象、构造函数与析构函数、继承与派生、冲突、支配规则和赋值兼容性与静态数据成员。第 11、12 章叙述了友元函数与运算符重载、多态性与虚函数、流类体系与文件操作。第 13 章介绍 C++综合编程实训，介绍使用结构体、链表、类和对象编写较复杂应用程序的方法。本书起点低，不要求学过其他程序设计语言，可作为程序设计的入门语言来学习。

本书可作为高职高专院校计算机、电子等专业的教材。

图书在版编目（CIP）数据

C++程序设计/周志德，侯正昌主编. —3 版. —北京：电子工业出版社，2011.11
高职高专计算机系列规划教材
ISBN 978-7-121-14813-2

Ⅰ. ①C… Ⅱ. ①周… ②侯… Ⅲ. ①C 语言－程序设计－高等职业教育－教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 210266 号

策划编辑：吕 迈
责任编辑：张 京
印 刷：
装 订：北京京师印务有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×1092 1/16 印张：23.5 字数：601.6 千字
印 次：2011 年 11 月第 1 次印刷
印 数：3 000 册 定价：40.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

C++是一种目前流行的面向对象程序设计语言，是学习 C#程序设计、Java 程序设计、数据结构、ASP.NET 程序设计、嵌入式程序设计课程的基础，是软件、计算机网络、计算机应用、电子信息、物联网应用技术等的专业基础课程。

本教材第一版于 2002 年出版。2005 年，对本书进行了修订。本教材于 2006 年被江苏省教育厅评为省精品教材。基于课程的重要性，作者在教学中采用三阶段教学法：第一阶段讲授“C++程序设计”，并辅以“计算方法与程序设计”选修课程；第二阶段开设为期一周的“C++课程设计”，第三阶段安排学生参加 C++等级考试培训，最后参加 C++二级考试。作者总结多年从教经验发现，计算机专业学生只有扎实掌握 C++语法、语义、算法基础，并通过 C++二级考试，才有可能学好 C#程序设计、Java 程序设计、ASP.NET 程序设计等后续课程，才能得到软件公司、网络公司、物联网公司的认可，并获得较好的职业岗位。

为此，作者对教材进行第 3 次修订，增加第 13 章“C++综合编程实训”；删除第 12 章中“二进制文件的使用”一节，并对流的格式控制方法做了适当修改。修订后的教材具有以下特点。

1．本教材将《C 语言程序设计》与《C++程序设计》综合为一本教材，用 C++语言来描述原先用 C 语言描述的内容，然后加上面向对象的程序设计内容。这样做的好处是：

（1）学生可以从数据类型、程序结构等基础内容开始由浅入深地进行学习，所以本教材起点低，可作为程序设计的入门教材来学习。

（2）由于将两门课综合为一门课，所以减少了总的教学时间，可在一学期内完成本门课的学习。

（3）可以使学生直接学习面向对象的程序设计方法。

2．针对高职高专学生的特点，本书尽可能使用通俗易懂的语言来叙述各章节内容，并尽可能使用典型例题来说明各章节知识点的概念与使用方法，力求将各章节的重点难点解释清楚，以求多数学生在教师讲课后能看懂教材，学会知识的应用。

3．由于描述 C++类与对象的程序段一般都较长，因此本书尽量用同一类型的例题介绍系列概念。例如，用描述学生成绩的类讲解类与对象的概念、定义及使用方法。用描述矩形的类介绍构造函数、拷贝构造函数、默认构造函数、析构函数等一系列的概念、定义及使用方法，以减少教师板书的工作量，提高课堂的讲课效率。

4．对学生比较难理解的内容，采用先通过例题分析，然后引出基本概念，给出定义格式、结论等正式内容的方式讲述。

5．高职高专学生在学习 C++程序设计时遇到的困难之一是：一方面要理解 C++中许多比较难的概念，另一方面又要理解复杂的算法。为了解决该问题，本教材在程序的算法上重点抓住常用的一些典型算法，如累加和、连乘积、最大值、最小值、平均值、排序，并将这些典型算法作为介绍各章节基本概念的例题，这样可减轻学生在理解算法上的负担，提高课堂的教学效果。一些有较复杂算法的例题只会出现在每章后的程序设计应用举例中。

6．每章前都有学习本章的目的、要求，每章后有本章小结，并配有一定量的习题，便

于教师教学和学生自学。各章内容充实，安排合理，衔接自然。

7．每章的小结中，列出了本章的重点和难点，便于教师实施教学和学生学习，提高学生学习效率。

8．每章后均有实验部分，其包含实验目的、实验要求、实验内容、实验提示等。将实验指导书综合在教材内，方便教师教学与学生做实验。

9．在部分章节中有习题课内容。习题课采用边小结内容边举例的方式进行。

10．对于某些算法（如三种排序算法），尽可能用简练的语句描述算法的核心含义，使读者易理解、易记忆。

11．第 5 章及以后的章节，凡是用到函数概念的知识点，如指针变量作为函数参数、返回指针值的函数、结构体数组作为函数参数、运算符重载函数，均以函数的定义、调用、参数传送三步进行描述。这样做的好处是：一方面，通过多次重复使学生能更好地掌握函数的定义、调用、参数传送的概念；另一方面，多次使用函数的定义、调用、参数传送三步方式更容易引入新知识点的概念。

12．增加了“C++综合编程实训”一章。通过实训，使学生了解使用 C++设计、开发简单信息管理系统的步骤和方法，掌握使用结构体、链表、类和对象编写应用程序，实现对数据进行插入、查询、修改、删除、排序和统计。

在本书编写的过程中，参考了目前国内比较优秀的有关 C++程序设计方面的书籍、资料，在此谨向有关作者表示感谢。对电子工业出版社吕迈编辑在修订工作中给予的大力支持表示衷心的感谢。

本书第 1～6 章由侯正昌编写，第 7～12 章由周志德编写，第 13 章由汪菊琴编写，全书由无锡职业技术学院周志德副教授统编，刘德强副教授审阅。其他执笔者有许敏、颜惠琴、王得燕、黄可望。

本书若有错误及不足之处，恳请读者给予指正。

编　者

2011 年 8 月

目　录

第 1 章　C++概述……1
1.1　C++的起源……1
1.2　C++的特点……1
1.3　C++程序的基本结构……2
1.4　C++上机操作……5
1.4.1　C++程序的开发步骤……5
1.4.2　C++程序上机操作方法……6
本章小结……9
习题 1……10
第 2 章　数据类型和表达式……11
2.1　数据类型……11
2.2　常量和变量……13
2.2.1　常量……13
2.2.2　变量……16
2.3　运算符和表达式……17
2.3.1　算术运算符和算术表达式……17
2.3.2　赋值运算符和赋值表达式……19
2.3.3　自增、自减运算符……20
2.3.4　关系运算符和关系表达式……21
2.3.5　逻辑运算符和逻辑表达式……22
2.3.6　逗号运算符和逗号表达式……24
2.3.7　复合赋值运算符……25
2.3.8　数据类型长度运算符（sizeof 运算符）……25
2.4　简单输入和输出……26
2.4.1　数据输出 cout……26
2.4.2　数据输入 cin……27
2.4.3　简单输入/输出格式控制……28
本章小结……29
习题 2……30
实验一……32
第 3 章　程序结构和流程控制语句……33
3.1　程序的三种基本结构和语句……33
3.1.1　程序的三种基本结构……33
3.1.2　C++程序的组成……34
3.1.3　C++程序的语句……35
3.2　分支语句……36
3.2.1　if 语句……36

3.2.2 条件运算符和条件表达式……41
3.2.3 switch 语句……42
3.3 循环语句……45
3.3.1 while 语句……45
3.3.2 do…while 语句……47
3.3.3 for 语句……49
3.3.4 三种循环语句的比较……52
3.3.5 循环语句的嵌套……52
3.4 控制执行顺序的语句……53
3.4.1 break 语句……53
3.4.2 continue 语句……54
3.4.3 语句标号和 goto 语句……55
3.4.4 exit()和 abort()函数……56
3.5 程序设计举例（习题课）……57
3.5.1 分支语句应用举例……57
3.5.2 循环语句应用举例……61
本章小结……65
习题 3……67
实验二……70
实验三……71
第 4 章 数组……73
4.1 数组的定义和使用……73
4.1.1 一维数组的定义和使用……73
4.1.2 二维数组的定义和使用……80
4.2 字符数组的定义和使用……84
4.2.1 字符数组和字符串……84
4.2.2 字符串处理函数……87
4.3 数组应用举例（习题课）……90
4.3.1 一维数组应用举例……90
4.3.2 二维数组应用举例……92
4.3.3 字符数组应用举例……95
本章小结……97
习题 4……98
实验四……100
实验五……101
第 5 章 函数……103
5.1 函数的定义和调用……103
5.1.1 函数的概念……103
5.1.2 函数的定义……104
5.1.3 函数的调用……105

5.1.4 实参与形参的数据传送 ……108
5.2 函数的嵌套调用和递归调用 ……109
5.2.1 函数的嵌套调用 ……109
5.2.2 函数的递归调用 ……110
5.3 数组作为函数参数（习题课） ……113
5.3.1 数组元素作为函数实参 ……113
5.3.2 数组名作为函数参数 ……114
5.4 变量的存储类型 ……118
5.4.1 作用域 ……118
5.4.2 局部变量与全局变量 ……120
5.4.3 动态变量与静态变量 ……121
5.4.4 变量的存储类型 ……121
5.5 内联函数 ……127
5.6 具有默认参数值的函数 ……128
5.7 函数的重载 ……129
本章小结 ……130
习题 5 ……133
实验六 ……137
第 6 章 编译预处理 …… 140
6.1 文件包含处理 ……140
6.2 宏定义 ……143
6.2.1 不带参数的宏定义 ……143
6.2.2 带参数的宏定义 ……145
6.3 条件编译 ……147
本章小结 ……150
习题 6 ……151
第 7 章 指针 …… 153
7.1 指针与指针变量 ……153
7.1.1 指针的概念 ……153
7.1.2 指针变量的定义与引用 ……153
7.1.3 指针变量的运算 ……155
7.2 指针与数组 ……160
7.2.1 一维数组与指针 ……160
7.2.2 二维数组与指针 ……161
7.2.3 字符串与指针 ……165
7.3 指针变量与数组作为函数参数（习题课） ……167
7.3.1 指针变量作为函数参数 ……167
7.3.2 数组与指针作为函数参数 ……169
7.4 指针数组和指向一维数组的指针变量 ……172
7.4.1 指针数组 ……172

7.4.2 指向一维数组的指针变量 ……174
7.5 返回指针值的函数与函数指针变量 ……176
7.5.1 返回指针值的函数 ……176
7.5.2 函数指针变量 ……179
7.6 new 和 delete 运算符 ……182
7.6.1 new 运算符 ……182
7.6.2 delete 运算符 ……183
7.6.3 使用 new 和 delete 运算符应注意的事项 ……184
7.7 引用类型变量和 const 类型的指针 ……185
7.7.1 引用类型变量的定义及使用 ……185
7.7.2 const 类型变量 ……186
本章小结 ……188
习题 7 ……192
实验七 ……194
实验八 ……195
第 8 章 枚举类型和结构体 ……197
8.1 枚举类型的定义及应用 ……197
8.1.1 枚举类型的定义 ……197
8.1.2 枚举类型变量的定义 ……198
8.1.3 枚举类型变量的引用 ……199
8.2 结构体的定义及应用 ……202
8.2.1 结构体的概念 ……202
8.2.2 结构体类型的定义 ……202
8.2.3 结构体变量的定义 ……203
8.2.4 结构体变量的引用 ……204
8.2.5 结构体变量与数组作为函数参数 ……206
8.3 链表 ……210
8.3.1 链表的概念 ……210
8.3.2 链表的基本操作 ……211
本章小结 ……220
习题 8 ……221
实验九 ……224
实验十 ……225
第 9 章 类和对象 ……227
9.1 概述 ……227
9.2 类与对象 ……229
9.2.1 类 ……229
9.2.2 对象 ……233
9.3 构造函数 ……236
9.3.1 构造函数的定义 ……236

9.3.2 用构造函数初始化对象的过程……238
9.3.3 默认构造函数……239
9.3.4 拷贝的构造函数……240
9.3.5 用 new 运算符动态定义对象……241
9.4 析构函数……243
9.4.1 定义析构函数……243
9.4.2 析构函数的调用……243
9.4.3 默认的析构函数……247
9.5 构造函数和对象成员……247
9.6 this 指针……250
本章小结……251
习题 9……253
实验十一……256
第 10 章 继承和派生类……258
10.1 继承与派生……258
10.1.1 继承与派生的基本概念……258
10.1.2 派生类的定义……260
10.1.3 派生类的构造函数与基类成员的初始化……262
10.2 冲突、支配规则和赋值兼容性……267
10.2.1 冲突……267
10.2.2 支配规则……269
10.2.3 赋值兼容规则……271
10.2.4 基类和对象成员的几点说明……271
10.3 虚基类……271
10.3.1 多重派生的基类拷贝……271
10.3.2 虚基类……273
10.4 静态数据成员……274
本章小结……277
习题 10……279
实验十二……282
第 11 章 友元与运算符重载……283
11.1 友元函数……283
11.1.1 定义普通函数为友元函数……283
11.1.2 友元注意事项……284
11.2 运算符重载……285
11.2.1 运算符重载的概念……285
11.2.2 二元运算符重载……285
11.2.3 一元运算符重载……290
11.2.4 字符串类运算符重载……298
11.3 多态性与虚函数……302

11.3.1 多态性技术……302
11.3.2 虚函数……302
11.3.3 纯虚函数……305
11.4 类与对象的特性……306
本章小结……307
习题 11……309
实验十三……311
第 12 章 流类体系与文件操作……313
12.1 流类体系……313
12.1.1 流……313
12.1.2 基本流类体系……314
12.1.3 标准输入/输出流……315
12.1.4 流的格式控制……316
12.1.5 数据输入/输出成员函数……321
12.2 文件操作……324
12.2.1 C++文件概述……324
12.2.2 C++的文件流类体系……324
12.2.3 文件的使用方法……325
12.2.4 文本文件的使用……329
本章小结……335
习题 12……337
实验十四……338
第 13 章 C++综合编程实训……340
13.1 系统需求分析……340
13.2 系统功能分析和模块设计……340
13.3 系统流程图与数据结构设计……340
13.3.1 系统流程图……340
13.3.2 数据结构设计……341
13.4 各功能模块程序设计……342
13.4.1 结构体……343
13.4.2 链表……347
13.4.3 类……349
13.5 学生成绩管理系统程序运行……350
13.6 其他系统设计要求……352
13.6.1 系统需求分析……353
13.6.2 系统功能分析和模块设计……353
13.6.3 系统流程图与数据结构设计……353
13.6.4 各功能模块程序设计……354
本章小结……355

附录 A　C++中的关键字 …… 356
附录 B　常用库函数 …… 358
附录 C　ASCII 码表 …… 361
附录 D　《C++程序设计》学时分配参考表 …… 362
参考文献 …… 363

第1章　C++概述

通过本章的学习，应了解C++的起源及其特点。掌握C++程序的基本结构，会编写极简单的C++程序。了解C++程序的开发步骤，熟悉Visual C++集成环境，初步掌握C++程序的上机操作方法。

1.1　C++的起源

C++是在C语言的基础上逐步发展和完善起来的，因此介绍C++之前不妨首先回顾一下C语言的发展。

1967年，Martin Richards为编写操作系统软件和编译程序开发了BCPL（Basic Combined Programming Language）；1970年，Ken Thompson在继承BCPL许多优点的基础上开发了实用的B语言；1972年，贝尔实验室的Dennis Ritchie在B语言的基础上，做了进一步的充实和完善，开发出了C语言。当时，设计C语言是为了编写UNIX操作系统，以后，C语言经过多次改进，逐渐开始流行。目前常用的C语言版本基本上都是以ANSI C为基础的。

C语言具有许多优点，如语言简洁灵活，运算符和数据结构丰富，具有结构化控制语句，程序执行效率高，同时具有高级语言和汇编语言的优点等。与其他高级语言相比，C语言具有可以直接访问物理地址的优点，与汇编语言相比又具有良好的可读性和可移植性。因此，C语言得到了极为广泛的应用。

随着C语言的推广，C语言存在的一些缺陷或不足也开始暴露出来，并受到大家的关注。例如C语言对数据类型检查的机制比较弱，缺少支持代码重用的结构；随着软件工程规模的扩大，难以适应开发特大型程序。同时，C语言毕竟是一种面向过程的编程语言，已经不能满足运用面向对象的方法开发软件的需要。C++便是在C语言基础上，为克服C语言本身存在的缺点，同时为支持面向对象的程序设计而研制出来的一种通用的程序设计语言，它是在1980年由贝尔实验室的Bjarne Stroustrup创建的。

研制C++的一个重要目标是使C++成为一个更好的C语言，所以C++根除了C语言中存在的问题；另一个重要目标就是面向对象的程序设计，因此在C++中引入了类的机制。最初的C++被称为“带类的C”，1983年正式命名为C++（C Plus Plus）。以后经过不断完善，形成了目前的C++。

当前运用较为广泛的C++有Microsoft公司的Visual C++（简称VC++）和Borland公司的Borland C++（简称BC++）。本书以Microsoft Visual C++ 6.0集成环境为背景介绍C++语言。

1.2　C++的特点

C++的主要特点表现在两个方面：一是全面兼容C语言，二是支持面向对象的程序设计方法。

（1）C++是一个更好的 C 语言，它保持了 C 语言的优点，大多数 C 程序代码略修改或不修改就可在 C++的集成环境下调试和运行。这对于继承和开发当前已在广泛使用的软件是非常重要的，可以节省大量的人力和物力。

（2）C++是一种面向对象的程序设计语言。它使得程序中各个模块的独立性更强，程序的可读性和可移植性更强，程序代码的结构更加合理，程序的扩充性更强。这对于设计、编制和调试一些大型软件尤为重要。

（3）C++集成环境不仅支持 C++程序的编译和调试，还支持 C 程序的编译和调试。通常，C++集成环境约定：当源程序文件的扩展名为.c 时，则为 C 程序；而当源程序文件的扩展名为.cpp 时，则为 C++程序。本书中，所有例题程序的文件扩展名均为.cpp。

（4）C++的语句非常简练，对语法限制比较宽松，因此 C++语法非常灵活。其优点是给用户编程带来书写上的方便。其缺点是由于编译时对语法限制比较宽松，许多逻辑上的错误不容易发现，给用户编程增加了难度。

1.3 C++程序的基本结构

为了说明 C++程序的基本结构，先举三个例题，然后通过三个例题引出 C++程序的基本结构。

【例 1.1】 文本的原样输出。文件名为 example1_1.cpp。

```
//文本原样输出程序
#include <iostream.h>
void main(void)
{    cout<<"Welcome to C++!\n";
}
```

该程序经编译和连接后，运行可执行程序时，在显示器上显示：

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

该程序中，main()表示主函数，每个 C++程序必须有且只能有一个主函数，C++程序总是从主函数开始执行。main()函数之前的 void 表示 main()函数没有返回值，main()函数后括号内的 void 表示 main()函数没有形式参数。花括号内的部分是函数体，函数体由语句组成，每个语句都以分号结束。cout 是 C++程序中的一个输出流，与符号“<<”结合使用可以输出常量、变量的值及原样输出双引号中的字符串。“\n”是换行符，即输出上述信息后换行。

程序中的#include 是 C++编译预处理中的文件包含命令，“iostream.h”是头文件，为了能使用输出流 cout 和输入流 cin，程序开头必须用#include 命令将文件 iostream.h 中的内容包含到本文件中来。

程序中以“//”开头的是注释，注释是对程序的说明，用来提高程序的可读性，可以放在程序的任何位置，对程序的编译和运行不起作用。

【例 1.2】 求两个整数的和。

```
/*求两个整数的和程序*/
#include <iostream.h>
```

```
void main(void)
{    int a,b,sum;                          //说明变量 a,b,sum 为整型数
     cout<<"Input a,b:";                   //显示提示信息
     cin>>a>>b;                            //从键盘上输入变量 a、b 的值
     sum=a+b;                              //求和
     cout<<"Sum="<<sum<<endl;              //输出结果
}
```

该程序经编译和连接后，运行可执行程序时，在显示器上显示：

```
Input a,b：3  5
Sum =8
```

该程序中的语句：int a,b,sum; 用来说明变量 a、b、sum 为 int（整型）变量。程序中的语句：sum=a+b；是一个赋值语句，表示将 a 和 b 的值相加，其结果送给变量 sum。在“/*”和“*/”之间的部分也表示注释。“endl”是换行符。

【例 1.3】 输入两个整数 a 和 b，用自定义函数 add()求两数之和。

```
#include <iostream.h>
int add(int x,int y)
{    int z;
     z=x+y;
     return z;
}
void main(void)
{    int a,b,sum;
     cout<<"Input a,b:";
     cin>>a>>b;
     sum=add(a,b);
     cout<<"Sum="<<sum<<endl;
}
```

该程序经编译和连接后，运行可执行程序时，在显示器上显示：

```
Input a,b：3  5
Sum=8
```

该程序由两个函数组成：主函数 main()和被调用函数 add()。函数 add()的作用是求 x 和 y 的和，并赋给 z，最后通过 return z 语句返回给主函数。主函数用来输入两个变量 a 和 b 的值，调用 add()函数将变量 a、b 的值传送给形参 x、y，再求两数和，并返回给 sum 输出结果。

通过例 1.3，可以归纳出 C++程序的基本结构如下。

（1）C++程序由包括 main()在内的一个或多个函数组成，函数是构成 C++程序的基本单位。其中名为 main()的函数称为主函数，可以将它放在程序的任何位置。但是，无论主函数放在程序的什么位置，一个 C++程序总是从主函数开始执行，由主函数来调用其他函数。所以，任何一个可运行的 C++程序必须有一个且只能有一个主函数。被调用的其他函数可以是系统提

供的库函数，也可以是用户自定义的函数。例如，例 1.3 中的 C++程序就是由主函数 main()和用户自定义函数 add()组成的。

（2）C++函数由函数说明与函数体两部分组成。

① 函数说明。函数说明由函数类型、函数名、函数参数（形式参数）及其类型组成。函数类型为函数返回值的类型。例如：

```
int add(int x,int y)
```

表示自定义了一个名为 add 的函数，函数值的类型为 int（整型），该函数有两个形式参数 x、y，其类型均为 int（整型）。

无返回值的函数是 void 类型（无值类型）。main()函数是一个特殊的函数，可看做是由操作系统调用的一个函数，其返回值是 void 类型或 int 类型。函数参数可以没有，但函数名后面的括号不能省略。

② 函数体。函数说明下面用花括号括起来的部分称为函数体。例如：

```
{   int z;                                  //变量定义
    z=x+y;                                  //执行语句
    return z;
}
```

如果一个函数内有多对花括号，则最外层的一对花括号为函数体的范围。通常函数体由变量定义和执行语句两部分组成。例如，int z；为变量定义，而 z=x+y；return z；为函数执行语句。在某些情况下可以没有变量定义，甚至可以既无变量定义又无执行语句（即空函数）。例如：

```
void dump(void )
 {   }
```

（3）C++中每一个语句和变量定义必须以分号结束。例如：

```
int z;   z=x+y ;
```

（4）C++程序的书写格式。C++程序的书写格式比较自由，一行内可以写多个语句（语句之间用“;”隔开），一个语句也可以分成几行来写。例如：

```
int add(int x,int y)
{   int z;   z=x+y;   return z;}   //将三条语句写在一行内
```

为了便于程序的阅读、修改和相互交流。程序的书写必须符合以下基本规则。

① 同层次语句必须从同一列开始书写，同层次的开花括号必须与对应的闭花括号在同一列上；

② 属于内一层次的语句，必须缩进几个字符，通常缩进两个、四个或八个字符的位置；

③ 任一函数的定义均从第一列开始书写。

（5）C++的输入/输出。C++没有专门的输入/输出语句，输入/输出操作是通过输入/输出流 cin 和 cout 来实现的。C++默认的标准输入设备是键盘，可形象地将 cin 理解为输入设备键盘。因此，cin>>a>>b；表示用键盘将数据输入变量 a 和 b。C++默认的标准输出设备是显示

器，可形象地将 cout 理解为输出设备显示器。因此，cout<<"Sum="<<sum<<endl；表示将字符串“Sum=”与变量 sum 的值输出到显示器上。

（6）C++严格区分字母的大小写。例如，int a,A；表示定义两个不同的变量 a、A。

（7）C++注释。在 C++程序的任何位置都可以插入注释信息。注释方法有两种：一种方法是用“/*”和“*/”把注释内容括起来，它可以用在程序中的任何位置。例如：

```
/*两个整数求和的程序*/
```

另一种方法是用两个连续的“/”字符，它表示从此开始到本行结束为注释内容。例如：

```
//说明变量 a,b,sum 为整型数
```

（8）编译预处理命令。以“#”开头的行称为编译预处理命令。例如，#include <iostream.h>表示本程序包含有头文件 iostream.h。

以上所述的有关函数、输入/输出流等概念将在以后的章节中详细介绍。C++程序的基本结构可用图 1.1 表示。

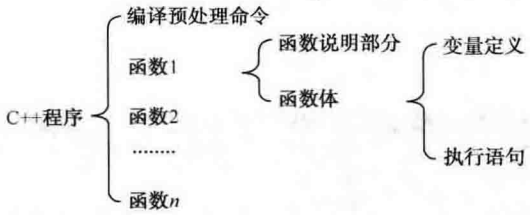

图 1.1　C++程序的基本结构

1.4　C++上机操作

1.4.1　C++程序的开发步骤

C++是一种编译性的语言，设计好一个 C++源程序后，需要经过编译、连接，生成可执行的程序文件，然后执行并调试程序。一个 C++程序的开发可分成如下几个步骤。

（1）分析问题。根据实际问题，分析需求，确定解决方法，并用适当的工具描述它。

（2）编辑程序。编写 C++源程序，并利用一个编辑器将源程序输入到计算机中的某一个文件中。源程序文件的扩展名为.cpp。

（3）编译程序。编译源程序，产生目标程序。目标程序文件的扩展名为.obj。

（4）连接程序。将一个或多个目标程序与库函数进行连接后，产生一个可执行文件。可执行文件的扩展名为.exe。

（5）运行调试程序。运行可执行文件，分析运行结果。若有错误则进行调试修改。

在编译、连接和运行程序过程中，都有可能出现错误，此时要修改源程序，并重复以上过程，直到得出正确的结果为止。