北京市高等教育精品教材立项项目

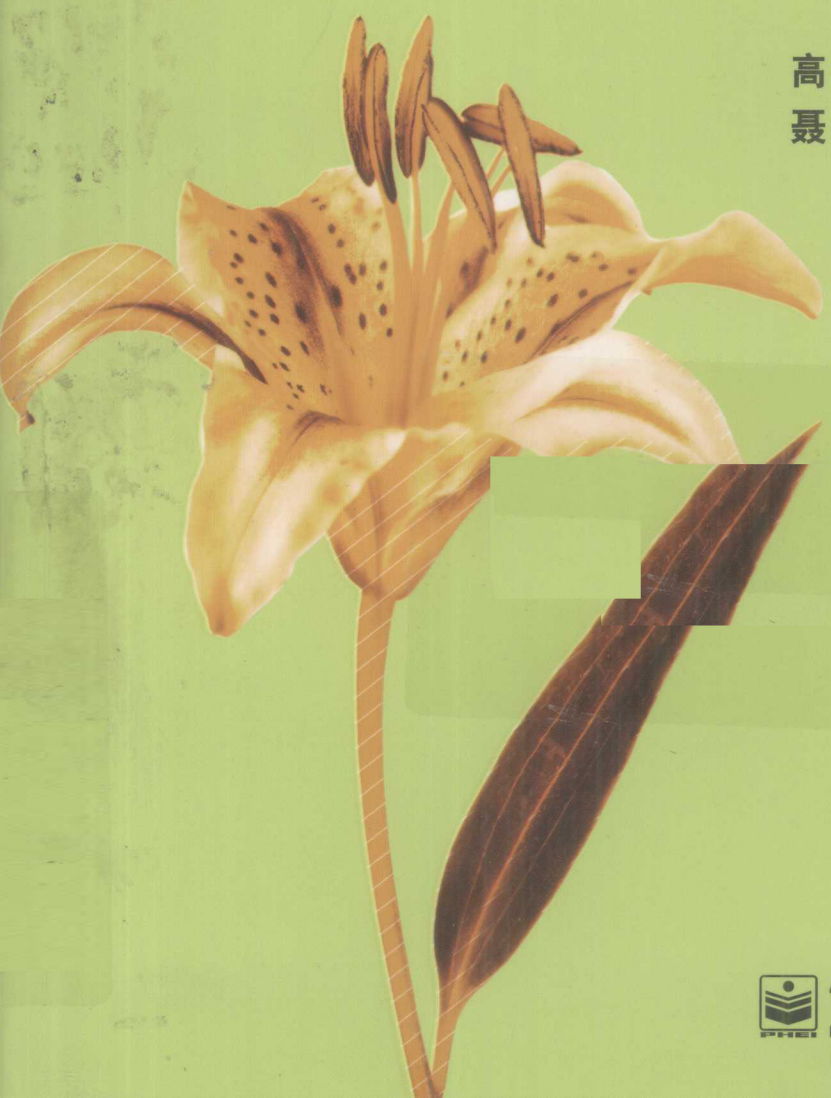
国家精品课程教材

高等学校工程创新型『十二五』规划计算机教材

C++与数据结构

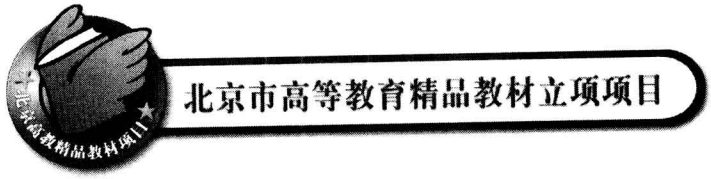
（第2版）

高　飞　薛艳明
聂　青　李慧芳　编著



Engineering Innovation

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



高等学校工程创新型“十二五”规划计算机教材

国家精品课程教材

C++与数据结构

(第2版)

高　飞　薛艳明
聂　青　李慧芳　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书是国家网络精品课程的教学成果，也是北京市高等教育精品教材立项项目，根据教育部计算机基础课程教学指导委员会《计算机基础课程教学基本要求》（试行）中，有关理工类专业“算法基础与程序设计”课程教学要求组织编写，内容由浅入深，循序渐进，案例丰富，通俗易懂，实用性强。全书分为C++程序设计基础和数据结构——面向对象方法与C++描述两篇，包括C++语言概述，数据类型与运算规则，数组与指针，函数，结构类型及其他构造类型，C++类及其对象的封装性，引用、友元和重载，继承与派生，多态性与虚函数，模板，数据结构基本概念，线性表，堆栈与队列，树与二叉树，图，查找与散列结构，排序共17章。本书各章节配有习题和实验训练题，方便实践教学，并为任课老师提供电子课件和示例源代码。

本书可作为高等学校信息类专业及其他相关专业本科生教科书，也可供从事程序设计的工程人员参考使用。

图书在版编目(CIP)数据

C++与数据结构/高飞等编著. —2版. —北京：电子工业出版社，2011.6
高等学校工程创新型“十二五”规划计算机教材
ISBN 978-7-121-13614-6

I. ①C… II. ①高… III. ①C语言－数据结构－程序设计－高等学校－教材 IV. ①TP312②TP311.12

中国版本图书馆CIP数据核字(2011)第091865号

策划编辑：童占梅
责任编辑：秦淑灵
印　　刷：北京丰源印刷厂
装　　订：三河市鹏成印业有限公司
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：22.25　字数：649千字
印　　次：2011年6月第1次印刷
印　　数：4000册　　定价：39.80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

本书是**北京市高等教育精品教材立项项目**，编写者是**国家级优秀教学团队**和**北京市优秀教学团队“计算机公共课教学团队”**的主要成员。教材的编写以教育部计算机基础课程教学指导委员会《计算机基础课程教学基本要求》（试行）中，有关理工类专业“算法基础与程序设计”课程教学要求为指导思想，结合教学改革的实践，总结了**国家级精品课程**和**北京市精品课程“数据结构与算法设计”**的建设经验，坚持以**培养学生解决实际问题的能力**为特色，以适应信息时代的人才培养模式。

C++是一种既支持面向过程程序设计又支持面向对象程序设计的混合型程序设计语言。它具有多C语言的向后兼容性，使很多已有的程序稍加修改就可以重用，许多有效的算法可以继续利用。同时还具有独特的面向对象特征，为面向对象技术提供全面支持。就软件而言，最重要的就是建立合理的软件体系结构、程序结构，以及设计有效的数据结构。因此，数据结构已经成为计算机程序设计的重要理论基础。

本书的写作特点

本书包括**C++程序设计基础**与**数据结构**（用面向对象方法与C++描述）两部分，力求使C++程序设计与数据结构的算法描述有机地融合在一起。C++是数据结构描述算法的一种理想工具。对于已经先修了“C语言程序设计”的读者，学习本书可以很容易地从C过渡到C++，花费较少的时间，就能掌握面向对象的程序设计思想与方法；对于没有先修过“C语言程序设计”的读者，则通过选学本书中的部分章节，即可掌握C++程序设计基础。本书用面向对象方法与C++来描述数据结构的算法，力求以算法为中介，使学生不但能提高对各种算法的理解与应用能力，而且还能提高程序设计的技能与素养，达到学习程序设计与数据结构融合共进的目的。

全书分为两篇，共17章。

上篇为第1～10章，它是C++程序设计基础。其中，第1～5章介绍C++简单程序设计，没有修过“C语言程序设计”的读者，则可通过选学这部分章节，掌握C++程序设计基础。第6～10章系统介绍C++的封装性、继承性、多态性，以及模板的概念与使用。

下篇为第11～17章，主要介绍面向对象方法与C++描述的数据结构，包括线性表、栈和队列、树与二叉树、图的概念、抽象类的描述及其不同存储表示的类的定义、实现和应用，并介绍了查找与散列的概念、散列函数的设计方法、解决冲突的多种策略、排序的概念、各种典型排序的算法和应用。

本书可作为高等学校信息类专业及其他相关专业本科生教材，也可供从事程序设计的工程人员参考使用。

本书的编排特点

本书每章安排有**习题和实验训练题**，方便实践教学。

对书中**重要内容采用黑体标记**。

本书**强调程序的可读性**。书中程序全部采用统一的设计风格。例如，类名、方法名和变量名的定义做到“望名知义”；语句的末尾或下一句的开头放上左大括号，而右大括号自成一行，并采用缩排格式组织程序代码；此外，对程序中的语句尽可能多地使用注释。

本书**包含大量的程序示例**，给出了运行结果。凡是程序开头带有程序名编号的程序，都是完整的程序，可以直接在计算机上编译运行。

本书采用醒目的标记来显示知识点，帮助读者尽快找到重要信息，这些标记有【程序解析】【程序运行结果】等。

课时分配

采用本书进行教学时，应该以课堂教学与上机实践相结合的方式进行。理想的教学时数与上机时数为 72+40 学时。经过对教材内容适当的削减和组织，本书同样适应 64+32 学时，以及 48+32 学时。具体学时安排可参考如下：

教 学 内 容	72+40 学时	64+32 学时	48+32 学时
第 1 章 C++语言概述	2+2	2	2
第 2 章 数据类型与运算规则	4+2	2	2
第 3 章 数组与指针	6+4	4+2	2+2
第 4 章 函数	4+4	2+2	2+2
第 5 章 结构类型及其他构造类型	4+2	2+2	2+2
第 6 章 C++类及其对象的封装性	4+2	4+2	4+2
第 7 章 引用、友元和重载	4+2	4+2	4+2
第 8 章 继承与派生	4+2	4+2	4+2
第 9 章 多态性与虚函数	4+2	4+2	4+2
第 10 章 模板	2+2	2+2	4+2
第 11 章 数据结构基本概念	2	2	2
第 12 章 线性表	6+2	6+2	4+2
第 13 章 堆栈与队列	6+4	6+4	6+4
第 14 章 树与二叉树	6+4	6+4	4+4
第 15 章 图	6+2	6+2	4+2
第 16 章 查找与散列结构	4+2	4+2	2+2
第 17 章 排序	4+2	4+2	2+2

教学支持

本书的电子教案可以在讲课时用多媒体投影演示，这可部分取代板书。教师不仅可以使用本教案，还可以方便地修改和重新组织其中的内容以适应自己的教学需要。使用本教案可以减少教师备课时编写教案的工作量，以及因板书所耗费的时间和精力，从而提高单位课时内的知识含量。

本书向使用本教材的教师**免费提供电子教案和程序示例源代码**。需要的教师可以直接登录华信教育资源网http://www.hxedu.com.cn免费注册下载。

本书由高飞设计总体构架。第 1，2，3，7，8，9 章由薛艳明编写，第 4，5，6，7 章由李慧芳编写，第 11，12，13 章由聂青编写，第 15，16，17 章由高飞编写。全书由高飞、薛艳明统稿、定稿。

在本书的编写过程中，电子工业出版社的编辑给予了很大的支持；高等教育分社童占梅副社长和秦淑灵编辑等对本书的编写提出了宝贵的意见，在此对她们的辛勤工作和热心帮助一并表示由衷的感谢！

由于计算机算法和程序设计技术发展迅速，作者水平有限及本书编辑仓促，书中的疏漏与不当之处在所难免，敬请广大读者和同仁不吝赐教，拨冗指正。

感谢您选用本书，欢迎您对本书内容提出意见和建议，作者 Email：gaofei@bit.edu.cn。

编著者

于北京理工大学

目　录

上篇　C++程序设计基础

下篇 数据结构——面向对象方法与 C++描述

上篇　C++程序设计基础

第1章　C++语言概述

1.1　C++语言简介

计算机的工作是通过程序来控制的，程序是指令的集合，指令则是计算机可以识别的命令。由于计算机硬件只能识别二进制指令构成的机器语言，所以在计算机诞生之初，只能使用机器指令编程，进而又出现了将机器指令映射为助记符的汇编语言。1954 年诞生了用于科学计算的高级语言 FORTRAN，之后又先后出现了 BASIC，ALGOL，COBOL 和 C 等多种高级语言。高级语言屏蔽了硬件的细节，提高了语言的抽象层次。采用高级语言编写的程序由具有一定含义的标识和容易理解的执行语句构成，为更多的人使用计算机提供了很大的方便。

在众多的计算机程序语言中，诞生于 20 世纪 70 年代的 C 语言在描述、开发系统软件和应用软件中具有重要地位。最初它是编写 UNIX 操作系统的工具，随着 UNIX 操作系统的广泛应用，C 语言迅速得到推广。后来经过不断完善改进，C 语言具有功能丰富、表达能力强、使用灵活方便、应用范围广、目标程序效率高、可移植性高等特点。更重要的是，C 语言是介于高级语言与低级语言之间的“中间语言”，既具有高级语言结构化与模块化的特点，又具有低级语言控制性与灵活性的特点。因此，C 语言被列为程序设计课程的首选语言，深刻影响了整整一代计算机工作者。

然而，C 语言也存在不足。例如，与其他高级语言相比，语法限制不严格。从应用的角度看，C 语言比其他高级语言较难掌握。而且，C 语言是面向过程的结构化和模块化的程序设计语言，当处理的问题比较复杂、程序规模较大时，就显得力不从心。为了更好地开发大型软件，20 世纪 80 年代提出了面向对象的程序设计方法，因此，需要设计出能够支持面向对象的程序设计方法的新语言。由于 C 语言应用广泛，深入人心，AT&TBell 实验室的 Bianrne Stroustrup 等人在 C 的基础上开发了它的增强版，这就是 C++语言。C++在保留 C 语言所有优点的基础上，增加了适用于面向对象的程序设计的“类”类型，因此，C++是一种既支持面向过程程序设计又支持面向对象程序设计的混合型程序语言。C++是 C 的超集，与 C 兼容。用 C 语言编写的程序基本上不加修改就可以用于 C++。C++不仅在 C 的基础上增加了面向对象的机制，而且对 C 语言的功能也做了不少扩充。

面向过程程序设计与面向对象程序设计是两种用途不同、互为补充的程序设计方法。面向对象程序中的成员函数就是用结构化程序设计方法实现的，任何程序的具体操作过程都是面向过程的。对于计算机硬件直接操作的问题，往往需要直接用面向过程的方法来解决。

C++是由 C 发展而来的，学习 C++程序设计方法，既要学会用 C&C++进行面向过程的结构化程序设计方法，也要学会用 C++进行面向对象的程序设计方法。

1.2　最简单的 C++程序

为了使读者了解什么是 C++程序，下面先介绍几个简单的程序。

【例 1.2.1】输出一行字符“This is a C++ program.”

程序如下：

```
#include <iostream>                                    //用 cout 输出时需要用此头文件
using namespace std;                                   //使用命名空间 std
int main()
{    cout <<" This is a C++ program.\n" ;              //输出一行信息
     return  0;
}
```

【程序运行结果】

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

【程序分析】

（1）在C++程序中，一般在主函数 main 前面加一个类型声明符 int，表示 main 函数的返回值为整型（标准 C++规定 main 函数必须声明为整型，即此主函数带回一个整型的函数值）。程序第 5 行的作用是向操作系统返回 0。如果程序不能正常执行，则会自动向操作系统返回一个非零值，一般为-1。

（2）在 C++程序中，既可以使用“/* ..... */”形式注释行，也可以使用以“//”开头的注释。从例 1.2.1 可以看出，以“//”开头的注释可以不单独占一行，它出现在一行中的语句之后。编译系统将“//”以后到本行末尾的所有字符都作为注释。应当注意的是，“//”是单行注释，不能跨行。C++的程序设计人员多愿意用这种注释方式，它比较灵活方便。

（3）在 C++程序中，一般用 cout 输出。cout 是由 c 和 out 两个单词组成的，它是 C++用于输出的语句。cout 实际上是 C++系统定义的对象名，称为输出流对象（对象的概念将在后面介绍）。为了便于理解，我们将 cout 和“<<”实现输出的语句简称为 cout 语句。“<<”是“插入运算符”，与 cout 配合使用。本例中它的作用是将运算符“<<”右侧双引号内的字符串“*This is a C++ program.*\n”插入到输出队列 cout 中（输出的队列也称“输出流”），C++系统将输出流 cout 的内容输出到系统指定的设备（一般为显示器）中。除了可以用 cout 进行输出外，在 C++中还可以用 printf 函数进行输出。

（4）使用 cout 需要用到头文件 iostream。程序的第 1 行“#include <iostream>”是一个预处理命令。文件 iostream 的内容是提供输入或输出时所需要的一些信息。iostream 是 i-o-stream 三个词的组合，可见它的含义是“输入输出流”。

（5）程序的第 2 行“using namespace std;”的意思是“使用命名空间 std”。C++标准库中的类和函数是在命名空间 std 中声明的，因此程序中如果要使用 C++标准库中的有关内容（此时需要用#include 命令行），就需要用“using namespace std;”语句来声明，表示要用到命名空间 std 中的内容。命名空间的概念暂不深究，只需知道，当程序有输入或输出时，必须使用“#include <iostream>”命令提供必要的信息，同时要用“using namespace std;”语句使程序能够使用这些信息，否则程序编译时将出错。请读者在写 C++程序时包含此两行语句。

需要注意的是，在 C 语言中，所有的头文件都带后缀.h（如 stdio.h），而按 C++标准要求，由系统提供的头文件不带后缀.h，但要与“using namespace std;”配合使用。用户自己编制的头文件可以有后缀.h。在 C++程序中，也可以使用 C 编译系统提供的带后缀的.h 的头文件，如“#include <math.h>”，此时不需使用“using namespace std;”语句。

【例 1.2.2】求 *a*、*b* 两个数之和。

```
#include <iostream >                                   /用 cout 输出时需要用此头文件
using namespace std;                                   //使用命名空间 std
int main()                                             //主函数首部
{    int a, b, sum;                                    //定义变量
     cin >> a >> b;                                    //输入语句
     sum = a + b;
     cout <<"a + b  =" << sum << endl;                 //输出语句
     return 0;
}
```

本程序的作用是求两个整数 a 与 b 之和 sum。“//”及后面的文字是程序的注释行，目的在于增加程序的可读性，对程序所完成的功能进行说明。在一行中，如果出现“//”，则从它开始到本行末尾之间的全部内容都作为注释。注释在编译之前会被预处理程序删除，对程序的可执行部分不产生任何影响。

第 1 行和第 2 行也可以用#include <iostream. h>一行来代替。第 4 行是声明部分，定义程序中用到的变量 a、b、sum。第 5 行是输入语句，cin 是 c 和 in 两个单词的组合，与 cout 类似，cin 是 C++系统定义的输入流对象。“>>”是“提取运算符”，与 cin 配合使用，其作用是从输入设备（如键盘）提取数据送到输入流 cin 中。用 cin 和“>>”实现输入的语句称为 cin 语句。在程序执行 cin 语句时，从键盘输入的第 1 个数据赋给 a，输入的第 2 个数据赋给 b。第 6 行将 a+b 的值赋给 sum。第 7 行先输出字符串“a+b =”，再输出变量 sum 的值。cout 语句中的 endl 是 C++输出时的控制符，作用是换行（endl 是 end line 的缩写，表示本行结束，与“\n”作用相同）。所以，在输出变量 sum 的值之后换行。

如果运行时从键盘输入

```
123  456 (回车)
```

则输出

```
a+b=579
```

【例 1.2.3】输入两个数 x 和 y，求两数中的较大者。

```
#include <iostream >                          //用 cout 输出时需要用此头文件
using namespace std;                          //使用命名空间 std
int main()                                    //主函数首部
{    int max( int x, int y );                 //对 max 函数进行声明
     int a, b, c;                             //定义变量
     cin >> a >> b;                           //输入语句
     c = max( a, b );                         //函数调用
     cout <<"max ="<< c << endl;              //输出语句
     return 0;
}
int max( int x, int y )
{    int z;
     return(  x > y ? x:y );
 }
```

【程序解析】本程序包括两个函数，即主函数 main 和被调用函数 *max*。*max* 函数的功能是求出 x 和 y 中的较大者，并用 return 语句返回给主调函数 main，通过函数名 *max* 带回到 main 函数的调用处。main 函数中 cin 的作用是输入 a、b 的值，main 函数的第 4 行调用 *max* 函数，调用时将实参 a、b 的值传给 *max* 中的形参 x、y，调用 *max* 后得到返回值传回主调函数，并赋给变量 c。

【程序运行结果】

```
18  25  (回车)      (输入 18 和 25 给 a、b)
max=25              (输出 c 的值)
```

注意：输入的两个数间用一个或多个空格间隔，而不能以逗号或其他符号间隔。例如，输入

```
18，25(回车)或  18；25 回车)
```

是错误的，它不能正确输入第 2 个变量的值，将使第 2 个变量有不可预测的值。

程序第 4 行是对 max 函数的声明，通知 C++编译系统 max 是一个函数，该函数有两个整型参数，返回值也是整型。编译系统会根据函数声明时的给定信息对函数调用的合法性进行检查，如果二者不匹配（参数个数或类型与声明时指定的不符），编译就会出错。

需要说明的是，以上几个例子是按 ANSI C++规定的语法编写的，由于 C++是从 C 发展而来的，为了与 C 兼容，C++保留了 C 语言的一些规定，其中之一是头文件的形式，C 语言的头文件以.h 为后缀，如 stdio.h、stdlib.h 等。

1.3 C++语言的基本组成

计算机程序设计语言与自然语言一样具有自己对字符、单词、特殊符号、语句、语法的使用规则。在 C++语言中，所涉及的规定主要包括基本字符集、标识符、语句与标准函数库等。

1.3.1 基本字符集

C++的基本字符集包括以下几部分。

（1）大小英文字母：A～Z，a～z；

（2）数字字符：0～9；

（3）运算符：+，－，*，/，%，=，<，>，<=，>=，！=，==，<<，>>，&，|，&&，||，^，～，()，[]，.，->，,，；；

（4）特殊字符：空格　_（下划线）及各种转义字符。

1.3.2 词法记号

C++的词法记号是通过其关键字和标识符体现的。

1. 关键字　C++预定义单词。用户定义的标识符不能和系统的关键字同名，以下是 48 个 C++的标准关键字：

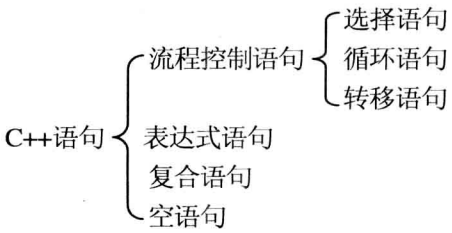
asm	auto	break	case	catch	char
class	const	continue	default	delete	do
double	else	enum	extern	float	for
frient	goto	if	inline	int	long
new	operator	private	protected	public	register
return	short	signed	sizeof	static	struct
switch	template	this	throw	try	typedef
union	unsigned	virtual	void	volatile	while

2. 标识符　程序员声明的单词，命名程序中的一些实体。

标识符的构成规则为，以大小写英文字母或下划线（_）开头，可以由大写英文字母、小写英文字母、下划线（_）或数字 0～9 组成。大写英文字母和小写英文字母表示不同的标识符。

1.3.3 语句

语句是组成程序的基本单位。所有程序设计语言都提供了满足编写程序要求的一系列语句，它们都有确定的形式和功能。C++语言的语句有以下几类：



1.3.4 标准函数库

C 语言程序中各种功能基本上都是由函数实现的。标准函数是由 C 编译系统提供的一些非常有用的功能函数。例如，C 语言没有输入输出语句，也没有直接处理字符串语句，而一般的 C 编译系统都提供了完成这些功能的函数。

调用函数库中的函数，需要程序的开头包含某些头文件。头文件中声明了很多函数原型，例如头文件“stdio.h”声明了 printf 函数原型。因此，在系统预定义函数 printf 函数时，必须在程序开始处加上#include <stdio.h>这样的预处理命令。有了包含命令，编译器将打开头文件“stdio.h”，并将其内容添加在该行位置，替换原先的包含命令。

C 语言在发展的过程中建立了功能丰富的函数库，由于 C++从 C 发展而来，因此，在 C++程序中可以使用 C 语言的函数库。在 C++中使用这些头文件有两种方法。

（1）遵循 C 语言规定的方法。C 语言的头文件以.h 为后缀，如 stdio.h、stdlib.h 等。由于 C 语言没有命名空间，头文件不可能存在于命名空间中，因此，C++程序中用到以.h 为后缀的头文件时，不必使用命名空间，只需像 C 语言程序那样，将头文件包含即可。例如“#include <stdio.h>”。

（2）用 C++的方法。按 C++标准要求，由系统提供的头文件不带后缀.h，但要与“using namespace std;”配合使用。例如：

```
# include <iostream>
  using namespace std;
```

为了与 C 语言的头文件既有联系又有区别，C++所用的头文件名是在 C 语言相应的头文件名（去掉后缀.h）最前面加字符 c。

用户自己编制的头文件可以有后缀.h。在 C++程序中，也可以使用 C 编译系统提供的带后缀.h 的头文件，如“#include <math.h>”，此时不须使用“using namespace std;”语句。例如，C 语言中的 stdio.h，在 C++中为 cstdio。C 语言中的 math.h，在 C++中为 cmath。

因为大多数 C++编译系统既提供了 C++的方法，又保留了 C 的方法，所以上述两种使用头文件的方法可以任选。

1.3.5 简单 C++程序结构

C++支持面向过程程序设计。面向过程的程序主要由顺序结构、选择结构和循环结构构成。

1．顺序结构程序设计

顺序结构程序执行顺序是程序书写的顺序。尽管一个 C++程序可以同时包含多种结构，但从宏观上来看，都可以看成顺序结构。

2．选择结构程序设计

在处理实际问题时，往往会面临许多判断，这时就需要根据不同的判断来执行不同的操作。为了解决此类问题，C++引进了选择结构。

C++中的 if 语句有两种形式：简单 if 语句和 if_else 语句。

（1）简单 if 语句

简单 if 语句：if（表达式）语句 1

功能：计算表达式的值，若为“真”，则执行语句 1；否则，跳过语句 1，执行 if 语句的下一条语句。

（2）if_else 语句

简单 if 语句只指出条件为“真”时做什么，而未指出条件为“假”时做什么。if_else 语句明确指出作为控制条件的表达式为“真”时做什么，为“假”时做什么。

if_else 语句的形式：

if（表达式）　语句 1
else　　语句 2

功能：计算表达式的值，若表达式的值为“真”，则执行语句 1，并跳过语句 2，继续执行 if_else 语句的下一条语句；若表达式的值为“假”，则跳过语句 1，执行语句 2，然后继续执行 if_else 语句的下一条语句。

（3）多分支结构

用嵌套 if 语句也能实现多分支结构程序，但分支较多时显得很烦琐，可读性较差。在 C 语言中，switch 语句专用于实现多分支结构程序，其特点是各分支清晰而直观。switch 语句调用形式：

```
switch（表达式）
{    case 常量表达式 1：语句 1
     case 常量表达式 2：语句 2
     …
     case 常量表达式 n：语句 n
     default：语句 n + 1
}
```

功能：首先计算表达式的值，然后依次与常量表达式 i（i = 1，2，…，n）比较，若表达式的值与常量表达式 j 相等，则从常量表达式 j 处开始执行（执行入口），直到 switch 语句结束。若所有的常量表达式 i（i = 1，2，…，n）的值均不等于表达式的值，则从 default 处开始执行。

3．循环结构

在许多问题中都需要用到循环控制，如求若干个数的和、排序等。很多高级语言都支持循环操作。一般来说，循环结构有以下几种：while 循环结构、do—while 循环结构、for 循环结构。

（1）用 while 语句设计“当型”循环结构程序

while 语句形式：

while（表达式）
**　循环体语句**

功能：首先计算表达式的值，若为“真”，则执行循环体语句，执行完毕再计算表达式的值，若仍为“真”，则重复执行循环体语句。直到表达式的值为“假”时，结束 while 语句的执行，继续执行 while 语句后面的语句。

（2）用 do_while 语句设计“直到型”循环结构程序

do_while 语句形式：

do　循环体语句
while（表达式）；

功能：首先执行循环体语句，然后检测循环控制条件表达式的值，若为“真”，则重复执行循环体语句，否则退出循环。

（3）用 for 语句设计“当型”循环结构程序

for 语句形式：

for（ 表达式 1；表达式 2；表达式 3 ）
**　循环体语句**

功能：首先计算表达式 1 的值；然后检测表达式 2 的值，若其值为“真”，则执行循环体语句，

执行完毕再计算表达式 3。然后，再测试表达式 2 的值是否为“真”，若为“真”，继续执行循环体语句……若为“假”，则终止循环。

1.4 C++程序的基本结构

通过上面的例子，我们已经初步了解了 C++程序的基本结构，归纳如下。

（1） 一个 C++程序是由一个程序单位或多个程序单位构成的。每一个程序单位作为一个文件。在程序编译时，编译系统分别对各个文件进行编译。因此，一个文件是一个编译单位。如上面简单的程序，只由一个程序单位（即一个文件）构成。

（2） 每个程序单位由若干个函数组成，因此 C++程序是由函数构成的。函数与函数之间是相对独立的、平行的，函数之间可以相互调用。在组成程序的若干个函数中，必须只有一个主函数 main()，它是程序执行的入口。

在一个程序单位中，可以包括：

① 预处理命令。C++程序的预处理命令以“#”开头，C++提供了 3 类预处理命令，即宏定义命令、文件包含命令和条件编译命令。

② 函数。函数是实现操作的部分，是程序中必须有的和最基本的组成部分。每一个程序则包括一个或多个函数，其中必须有一个（而且只能有一个）主函数（main 函数）。

（3）一个函数由如下两部分组成：

① 函数首部，即函数的第一行。包括函数名、函数类型、函数属性、函数参数（形参）名和函数类型。

例如，例 1.2.1 中 main 函数的首部为

```
int        main   (  )
 ↓          ↓
函数类型   函数名
```

一个函数名后面必须跟一对圆括号，括号内的函数参数可以省略。

② 函数体，即函数首部下面花括号内的部分。如果在一个函数中有多个花括号，则最外层的一对{ }为函数体的范围。

函数体一般包括如下两部分：

- 声明部分。包括对本函数中所用到的变量类型、函数的声明。
- 执行部分。由若干个执行语句组成，用来进行有关的操作，以实现函数的功能。

当然，在某些情况下也可以没有声明部分，甚至可以既没有声明部分，也没有执行部分，如

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

它是一个空函数，不执行任何功能，但这是合法的。

（4）语句是组成程序的基本单元，函数是由若干条语句组成的（空函数除外）。语句是由单词组成的，单词间以空格符分隔，语句以分号结束。一条语句结束时用分号，没有结束时不用分号。C++中的语句包括两类，一类是声明语句，如“int a，b;”，用来向编译系统通知某些信息（如函数和变量的声明和定义），但它并不引起实际的操作，是**非执行语句**；另一类是**执行语句，用来实现指定的操作**。C++对每一种语句赋予一种特定的功能。语句是实现操作的基本成分。C++语句必须以分号结束，如“c = a + b;”，分号是语句的一个组成部分，没有分号则不是语句。

（5）一个 C++程序总是从 main 函数开始执行的，不论 main 函数在程序中的位置如何（main 函数可以放在程序文件的最前头，也可以放在程序文件的最后，或者在一些函数之前，在另一些函数之后）。

（6） C++程序书写格式自由，基本原则如下：

① 一行内可以写多条语句，一条语句也可以分写在多行。短语句可以在一行写多条，长语句可以一条写在多行。C++程序没有行号，也不像 FORTRAN 或 COBOL 语言那样严格规定书写格式（语句必须从某一列开始书写）。

② 一条语句分行的原则是不能将单词分开，用双引号引用的字符串最好也不要分开。

③ 花括号{ }的书写方法较多，常用的是每个花括号占一行，并与使用花括号的语句对齐，花括号内的语句采用缩格书写的方式。

④ C++程序书写时要尽量提高可读性，为此，采用适当的书写形式是很重要的。首先，表示同一类内容的语句行要对齐，例如，一个循环的循环体中各语句要对齐，同一个 if 语句中 if 体内的若干语句或 else 体内的若干语句都要对齐。其次，一个好的、有使用价值的源程序应该加上必要的注释。在前面的程序中用“//”作为注释行的标志，在一行中从“//”开始到本行末的全部内容作为注释。注释是给人看的，不是给计算机看的，在程序编译和运行时不起作用。C++还保留了 C 的注释形式，可以用“/*.....*/”对 C++程序中的任何部分做注释。在“/*”和“*/”之间的全部内容作为注释，以下两行等价：

```
// Hello! Welcome to C++program.
/* Hello! Welcome to C++program.*/
```

但是，用“//”做注释，有效范围只有一行，即本行有效，不能跨行。如果注释的内容较多，需要用多个注释行，每行用一个“//”开头。而用“/*.....*/”做注释时有效范围为多行。只要在开始处有一个“/*”，在最后一行结束处有一个“*/”即可。因此，一般习惯是，内容较少的简单注释常用“//”，内容较长的注释常用“/*.....*/”。

1.5 C++程序设计的编写与实现

前面已经看到一些用 C++语言编写的程序。但是，写出程序并上机调试运行通过，才能得到最终的结果。一个程序从编写到最终得到运行结果需要经历如下步骤。

（1）用 C++语言编写程序

程序就是一组计算机系统能够识别和执行的指令。每条指令使计算机执行特定的操作。用高级语言编写的程序称为源程序（Source Program）。C++的源程序以.cpp 为后缀，cpp 是 c plus plus 的缩写。

（2）对源程序进行编译

从根本上说，计算机只能识别和执行由 0 和 1 组成的二进制指令，而不能识别和执行用高级语言编写的指令。为了使计算机能执行高级语言源程序，必须先用“编译器（Complier）”软件（也称编译程序或编译系统），把源程序翻译成二进制形式的“目标程序（Object Program）”。

编译是以源程序文件为单位分别编译的，每一个程序单位组成一个源程序文件，如果有多个源程序单位，系统将分别把它们编译成多个目标程序。目标程序一般以.obj（Object 的缩写）作为后缀。编译的作用是对源程序进行词法检查和语法检查。词法检查是检查源程序中的单词拼写是否有错，语法检查是根据源程序的上下文检查程序的语法是否有错。编译时对文件中的全部内容进行检查，并在编译结束时显示所有的编译出错信息。一般编译系统给出的出错信息有两种，一种是错误（Error），另一种是警告（Warning）。warning 是一些不影响运行的轻微错误。如果编译时检查出 error 错误，就不能生成目标程序了，而必须回到步骤（1），修正错误，重新编译。

（3）连接程序

在对编译时检查出的错误全部改正并编译通过后，会得到一个或多个目标文件。此时，需要利用系统提供的“连接程序（Linker）”将程序的所有目标文件、编译系统的库文件及系统提供的其他信息连接起来，形成一个可执行的二进制文件（后缀是.exe）。