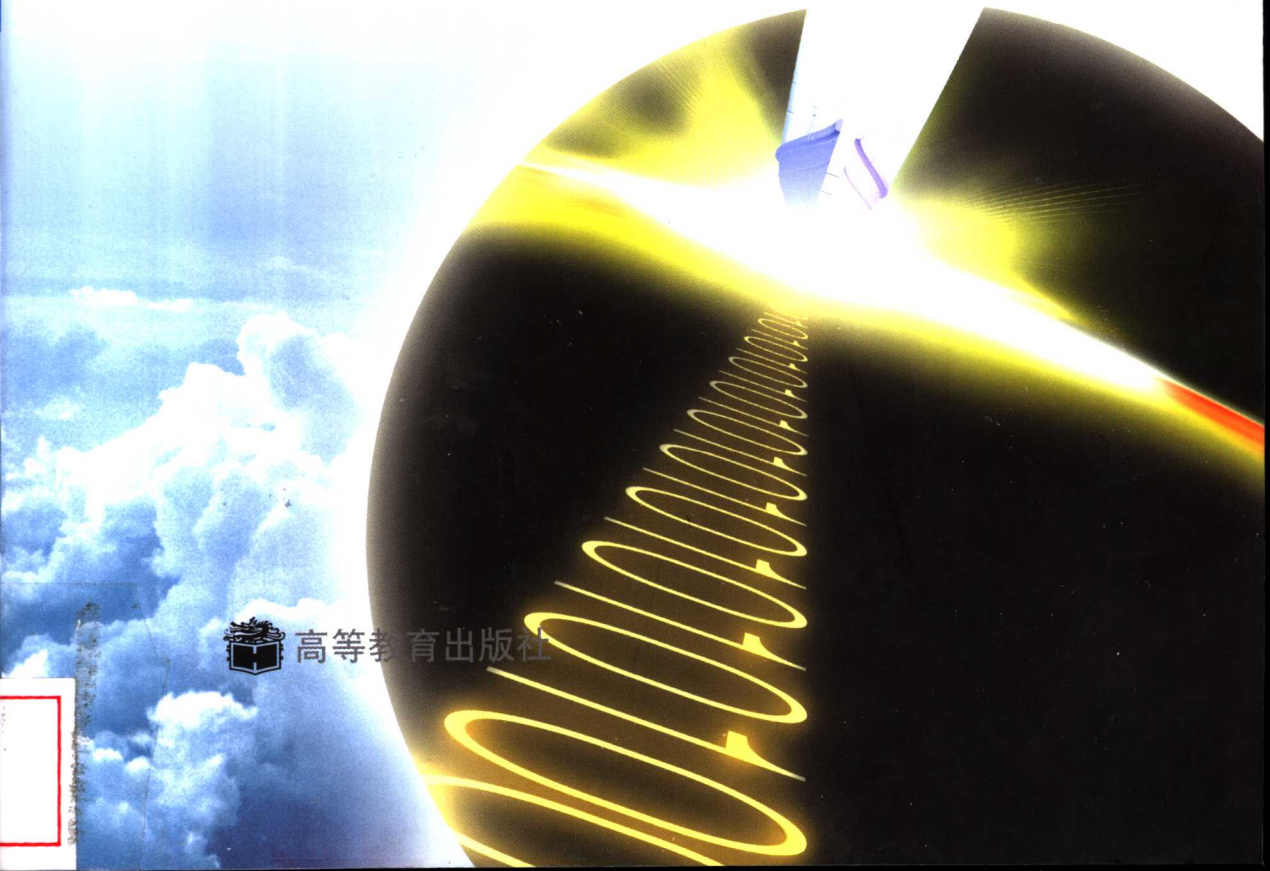
教育科学“十五”国家规划课题研究成果

实用 C++ 教程

马智娴　徐克奇　荣　祺　邹金安



高等教育出版社

教育科学“十五”国家规划课题研究

实用 C++ 教程

马智娴　徐克奇　荣　祺　邹金安

高等教育出版社

内容提要

本教材分三部分，第一部分 C++ 语言基础，第二部分面向对象的程序设计，第三部分使用 MFC 开发 Windows 程序。内容安排由浅入深，循序渐进，便于初学者系统地学习。

本教材通俗易懂，重点突出，强调面向应用，尽量避免深奥的理论，用浅显易懂和简明的例子说明 C++ 各种复杂的技术概念。

本教材可作为应用型本、专科院校学生的教材，也可供计算机爱好者自学。本书并配有配套的实验教程，可使读者通过具体实验更深入地理解所学的内容。

图书在版编目(CIP)数据

实用 C++ 教程/马智娴等. —北京：高等教育出版社，2004.11

ISBN 7-04-015627-X

Ⅰ.实… Ⅱ.马… Ⅲ.C 语言-程序设计-高等学校-教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2004)第 106288 号

策划编辑 付 欣 **责任编辑** 付 欣 **市场策划** 刘 茜
封面设计 刘晓翔 **责任印制** 陈伟光

出版发行	高等教育出版社	**购书热线**	010-64054588
社 址	北京市西城区德外大街 4 号	**免费咨询**	800-810-0598
邮政编码	100011	**网 址**	http://www.hep.edu.cn
总 机	010-58581000		http://www.hep.com.cn
经 销	新华书店北京发行所		
印 刷	北京民族印刷厂		
开 本	787×960 1/16	**版 次**	2004 年 11 月第 1 版
印 张	22	**印 次**	2004 年 11 月第 1 次印刷
字 数	410 000	**定 价**	25.30 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

物料号：15627-00

总　序

为了更好地适应当前我国高等教育跨越式发展需要，满足我国高校从精英教育向大众化教育的重大转移阶段中社会对高校应用型人才培养的各类要求，探索和建立我国高等学校应用型本科人才培养体系，全国高等学校教学研究中心（以下简称“教研中心”）在承担全国教育科学“十五”国家规划课题——“21世纪中国高等教育人才培养体系的创新与实践”研究工作的基础上，组织全国100余所以培养应用型人才为主的高等院校，进行其子项目课题——“21世纪中国高等学校应用型人才培养体系的创新与实践”的研究与探索，在高等院校应用型人才培养的教学内容、课程体系研究等方面取得了标志性成果，并在高等教育出版社的支持和配合下，推出了一批适应应用型人才培养需要的立体化教材，冠以“教育科学‘十五’国家规划课题研究成果”。

2002年11月，教研中心在南京工程学院组织召开了“21世纪中国高等学校应用型人才培养体系的创新与实践”课题立项研讨会。会议确定由教研中心组织国家级课题立项，为参加立项研究的高等院校搭建高起点的研究平台，整体设计立项研究计划，明确目标。课题立项采用整体规划、分步实施、滚动立项的方式，分期分批启动立项研究计划。为了确保课题立项目标的实现，组建了“21世纪中国高等学校应用型人才培养体系的创新与实践”课题领导小组（亦为高校应用型人才立体化教材建设领导小组）。会后，教研中心组织了首批课题立项申报，有63所高校申报了近450项课题。2003年1月，在黑龙江工程学院进行了项目评审，经过课题领导小组严格的把关，确定了首批9项子课题的牵头学校、主持学校和参加学校。2003年3月至4月，各子课题相继召开了工作会议，交流了各校教学改革的情况和面临的具体问题，确定了项目分工，并全面开始研究工作。计划先集中力量，用两年时间形成一批有关人才培养模式、培养目标、教学内容和课程体系等理论研究成果报告和在研究报告基础上同步组织建设的反映应用型人才培养特色的立体化系列教材。

与过去立项研究不同的是，“21世纪中国高等学校应用型人才培养体系的创新与实践”课题研究在审视、选择、消化与吸收多年来已有应用型人才培养探索与实践成果基础上，紧密结合经济全球化时代高校应用型人才培养工作的实际需要，努力实践，大胆创新，采取边研究、边探索、边实践的方式，推进高校应用型本科人才培养工作，突出重点目标，并不断取得标志性的阶段成果。

教材建设作为保证和提高教学质量的重要支柱和基础，作为体现教学内容

和教学方法的知识载体，在当前培养应用型人才中的作用是显而易见的。探索、建设适应新世纪我国高校应用型人才培养体系需要的教材体系已成为当前我国高校教学改革和教材建设工作面临的十分重要的任务。目前，教材建设工作存在的问题不容忽视，适用于应用型人才培养的优秀教材还较少，大部分国家级教材对一般院校，尤其是新办本科院校来说，起点较高，难度较大，内容较多，难以适应一般院校的教学需要。因此，在课题研究过程中，各课题组充分吸收已有的优秀教学改革成果，并和教学实际结合起来，认真讨论和研究教学内容和课程体系的改革，组织一批学术水平较高、教学经验较丰富、实践能力较强的教师，编写出一批以公共基础课和专业、技术基础课为主的有特色、适用性强的教材及相应的教学辅导书、电子教案，以满足高等学校应用型人才培养的需要。

我们相信，随着我国高等教育的发展和高校教学改革的不断深入，特别是随着教育部即将启动的“高等学校教学质量和教学改革工程”的实施，具有示范性和适应应用型人才培养的精品课程教材必将进一步促进我国高校教学质量的提高。

全国高等学校教学研究中心

2003 年 4 月

前　言

随着计算机技术的进步与普及，人们对计算机的兴趣与日俱增，计算机技术已经渗透到了各行各业。正因为如此，社会上迫切需要大量掌握计算机技术的人才。为了满足人们的需要，我们编写了这本 C++ 的入门教材。

本书主要是通过易学的形式来学习 C++。本书分三部分，分别是C++ 语言基础、面向对象的程序设计和使用 MFC 开发 Windows 程序。本书内容由浅入深，循序渐进，便于初学者系统地学习。

C++ 课程是一门对实践性要求较高的课程。本书的特点是：通俗易懂，重点突出，强调面向应用，尽量避免深奥的理论，用浅显易懂和简明的例子说明 C++ 各种复杂的技术概念。为此，每一章都提供了一系列的实例，用这些实例演示了相关的概念，并通过本书配套的实验教程，使读者通过具体的实验来更深入地理解所学的内容。

总之，编写本书的目的就是帮助和指导读者学会 C++ 的基本概念和实际使用技术，带你进入 C++ 的世界。希望通过本书的学习，对你有所帮助。

书中所有的例题都已在中文 Windows 系统、Microsoft Visual C++ 6.0 中文版环境下编译通过。

本书的第一部分由荣祺编写，第二部分由徐克奇编写，第三部分由马智娴、邹金安编写，马智娴统编全书。特请周钦铭审阅了全书，在此表示感谢。

书中如有不妥之处，欢迎读者批评指正。

编　者

2004 年 7 月

目　录

第一部分　C++语言基础

第二部分 面向对象程序设计

第三部分　使用 MFC 开发 Windows 程序

第一部分

C++语言基础

第1章　C++语言基础知识

本章将讨论C++语言的基础知识，包括C++语言的程序结构、数据类型、运算符号与表达式、控制流语句和基本输入/输出等方面内容。

1.1　C++程序结构

1.1.1　简单的C++程序介绍

首先，通过一个最简单的C++程序来讨论C++程序的结构：

[例1-1]　什么都不做的C++程序

```
void main()
{
}
```

说明：

输入并运行这个程序，可以发现什么运行结果都没有。虽然这个程序什么事情都不做，但这的确是一个完整的C++程序。以下是一些基本概念的说明。

(1) 注释：在这个程序中，可以看到 /* 和 */这样两个符号，这两个符号在C++中通常必须同时出现，称为注释符号。在注释符号之间，编程者可以加入一行或多行文字，用于说明程序的目的或程序中特定语句的作用，以便以后阅读程序。虽然从程序运行角度出发，注释的确是可有可无的，但从良好的程序设计风格角度出发，注释还是需要的，毕竟它可以帮助你或别人更好地阅读和理解代码，提高效率。

(2) 函数：在这个什么都不做的程序中，还是键入了一个函数，也就是

```
void main() {}
```

观察它的形式，可以发现是由 void、main、()和{}组成，其中，void 表示这个函数不返回任何类型的数据；main 表示函数的名称；()表示函数参数列表，尽管当前写的函数没有任何参数，但还是需要写个空括号；最后的{}表示函数体，其中包括实现函数功能的程序语句，因为这个程序什么都不做，所以这里也仅仅只是写了一对大括号。

关于函数的理解，可借助数学中的函数，如 $y = \sin(x)$，sin 表示正弦函数，当 $x = 30°$时，$y = 0.5$。其中，x 称为自变量，y 之所以等于 0.5，是因为将 sin(30°)计算出的结果后赋值给了 y；如果用C++的角度去看，sin 称为函数名，()表示参数列表，写在()中的 x 为参数，而返回的计算结果称为返回值，至于正弦函数的功能是如何实现的，那就需要在花括号中添加具体的实现程序语句了，称为函数体。

(3) 程序入口：或许这里有个疑问，为什么用 main 名称作为函数名而不是别的什么字符组合，这是因为，在标准C++程序设计中①，固定使用这个名称来表示程序开始执行的起点，规定以 main 为函数名的函数为主函数。在一个程序中，必须有而且只能有一个主函数，原因很简单，计算机执行C++程序是通过 main()来确定执行的起点的，因为被执行程序的起点只能有一个，所以 main 函数必须有且只能有一个，故此，main 函数通常也被认为是特殊函数。

至此，你或许会得出的第一个结论是：C++程序是由函数构成的，函数构成了程序的主体；而在程序中必然有个名为 main()的主函数。而另外一个结论是如果想让程序做点什么，那必须要在函数体中写点什么。下面的例程证实了这一想法是正确的。

[例 1-2] 实现用户输入的两个整数相加的C++程序

```
#include "IOStream.h"
void main()
{
    //变量声明部分
    int i,j;
    int sum;

    //用户输入部分
    cout << "输入第一个整数：";
    cin >> i;
    cout << "输入第二个整数：";
    cin >> j;

    //处理部分
    sum = i + j;

    //输出部分
    cout << "运行结果：" << i << " + " << j << "= " << sum << endl;
}
```

① 利用C++ WIN32 程序设计时候，通常是 WinMain()作为程序的入口。

程序运行结果：

输入第一个整数：　*1*[①]
输入第二个整数：　*2*
运行结果：1 + 2 = 3

这个例程稍微复杂一点，下面是相关的说明：

① 预处理命令：可以看出例 1 - 2 和例 1 - 1 的区别除了在 main 函数的函数体中多了些代码，在 main 函数前面多了一条语句 # include " IOStream.h " ，这句以“ # ”开头的语句，在C++ 中被称为预处理命令，在C++ 中，预处理命令包括：宏定义命令、文件包含命令和条件编译命令，而像这里以 # include 开始的表示预处理命令中的文件包含命令，表示本程序需要将 IOStream.h 文件中的程序代码包含进来。

那为什么要包含这个文件呢？因为在 main( ) 的函数体包含的代码中用到了 cin 来接收用户输入的数据，并用 cout 来输出运算的结果，之所以可以使用这些符号来实现对应的功能，是因为实现这些功能的相关代码在 IOStream.h 文件中已经写好了，而且通过 # include " IOStream.h " 命令将它们给包含进来了。

总而言之，C++ 中自带了很多文件，这些文件中已经定义了许多特定的功能，分别用于实现不同方面的功能，如输入/输出、文件处理等，所以当使用到这些C++ 已有功能的时候，只需要包含这些特定的文件。通常，这些文件称为头文件或函数库文件，在本书附录或其他C++ 技术文档中，可以很方便地了解相关头文件中定义的功能。如：在例程中包含的 IOStream.h 头文件，定义的是关于C++ 输入和输出方面的基本功能，是最常用的函数库文件之一。

② 变量声明：仔细考察 main 函数体内的代码组成，变量声明部分中的语句是诸如 int i,j 和 int sum;之类的形式，在C++ 中，int 表示整数，而 int 后面的 i、j 和 sum 表示变量名称。需要声明变量的原因是，用户输入的数据或程序中计算的结果等都需要特定的空间加以保存，所以为了有合适大小的存储空间，需要声明变量。变量相当是一个用于存放数据的容器，因为是一个容器，所以有容量限制。在C++ 中，用于表达数据容量的字符组合，称为数据类型，如本例程中的 int，在后面的论述中，还会看到更多的数据类型，如 long、double 等。

③ 基本输入和输出：在例程中，使用了 cin 和 cout 来完成对应的输入和输出功能，可以发现 cin 后面跟随的是 >> 符号(两个连写的大于符号)，表示将输入的数据传向某个变量，如：cin >> i 表示将用户输入的数据存放到 i 中；而 cout 后面跟随的是 << 符号(两个连写的小于符号)，表示将后面变量中存储的数据或括在一对双引号中特定的文字串输出，如：cout << " 运行结果： " << i <<

① 程序运行结果中斜体表示用户输入部分，以下同。

" + " << j << "= " << sum << endl 表示将 i 中存储的数据和 j 中存储的数据和 " " 中特定的文字串依次输出，得到的结果就是“运行结果：1 + 2 = 3”，符号 endl 的功能之一表示回车换行，也就相当于在键盘上敲个回车的效果。

需要说明的是，在 cin 的使用中，也可以跟多个 >>，如：cin >> i >> j，表示将用户输入的数据依次存放到变量 i 和 j 中。当然，必须在输入数据的时候用空格或回车来隔开输入的多个数据。

回顾关于 cin 和 cout 的论述，可以发现 cin 像一个数据源，不断将数据根据箭头 >> 指示的方向依次储存到后继的变量中；而 cout 相当一个接收器，数据根据 << 方向不断地流向它，最终被输出。数据无论是在 cin 或 cout 中，都根据特定的方向进行流动，故此通常在C++ 中这种输入/输出操作称为“流”操作。关于流，在后面章节中会有更加详细的论述。

至此，可以看出如果希望程序能做点什么，就需要在函数体内写些代码，写代码的时候或许需要包含一些特定的头文件，来调用C++ 中已提供的一些功能，从而简化编程。而函数体内的代码，通常包括变量声明、输入、处理和输出结果几个部分，需要指出的是并不是所有的程序代码都必须要这几个部分，而是根据实际需要而定。

在前面两个例程中，都只有一个函数，那就是 main 函数。其实在C++ 程序中，更常见的是通过多个函数来完成程序功能，见如下例程。

[例 1-3] 实现求两个整数中较大值的C++ 程序

```
#include "IOStream.h"
int max(int ii,int jj)
{
   //比较两个整型数的大小,并返回较大数
   if( ii >= jj )
      return ii;
   else
      return jj;
}

void main()
{
   //变量声明部分
   int i = 1,j = 2;
   int m;
   //处理部分
   m = max(i,j);

   //输出部分
```

```
    cout << i << "和" << j << "中较大值是:" << m << endl;
}
```

程序运行结果:

```
1 和 2 中较大值是:2
```

说明:

① 用户函数:考察上面例程的代码可见,多了一个 max 函数,用于比较两个整数的大小,它的形式是

```
int max(int ii,int jj ){ … }
```

和 main 函数形式非常相似,只是这个函数前面不是 void,而是 int,这表示 max 函数有一个 int 类型的返回值,也就是说,这个函数执行后返回了一个 int 类型的整数。考虑到这个函数需要返回被比较的两个整数中较大的那个,所以需要这样一个返回值就非常容易理解了;其参数区,有两个参数,分别是 ii 和 jj,注意到两个参数的前面也有数据类型说明,是因为需要说明如何正确使用这个函数,也就是说,在使用这个函数的时候,必须告诉这个函数两个数,而且必须是 int 类型的参数才可以正确使用 max 函数。另外一个原因是因为这里列出的参数是需要作为变量在 max 函数中使用的。

在函数体中出现的代码,是一些实现比较操作的语句,目前无须关注这些代码的具体细节,后面章节会有进一步说明。用户根据实际的需要,定义了 max 函数,并在其中编写代码实现了比较两个整数大小的功能,这种函数通常称为用户函数,也就是用户为实现特定功能而编写的函数。

变量初始化:在这个例程中,并没有使用 cin 来给变量赋值,而是写了如下的代码:int i = 1,j = 2; 即在声明变量类型的同时对变量进行了赋值。相当于在申请了两个变量空间的同时,在变量空间中放入指定的数据。注意到这种声明变量的同时进行赋值的过程,称为变量初始化。在特定场合,变量初始化和变量声明后,再在其他语句中用"="赋值是有区别的,这在后面的章节中可以看到,希望读者注意。

② 函数调用过程:如果单从 main 函数体中的代码来看,处理部分只有一句:m = max(i,j);就实现了求较大值的目的,具体比较操作是在 max 函数中实现的,或者说在 main 函数中调用了 max 函数的功能,类似这样的过程称为函数调用,其中,main 函数称为主调函数,而 max 函数称为被调函数。在C++ 程序设计中,正是利用函数调用将程序中的多个函数有机地组织起来。

在 main 函数中调用 max 函数的过程是:根据 max 函数定义参数的类型和数量,定义了 i、j 两个 int 类型变量,并分别赋值 1 和 2;并定义了一个 int 类型的变量 m,变量 m 并没有初始化。考虑一下,为什么? 再看一下 max 函数的定义,就非常清楚了,因为 max 函数有两个 int 类型参数的同时,还有一个 int 类型的返回