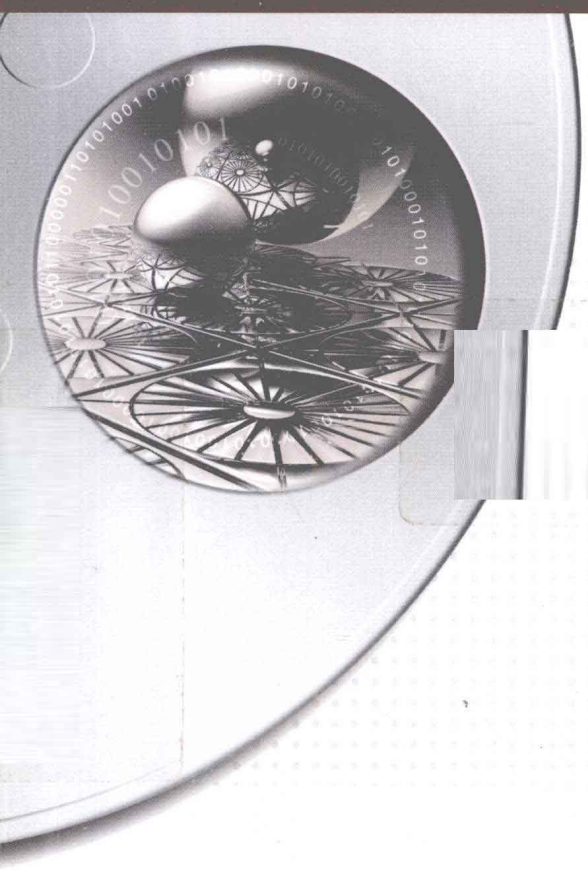
普通高等教育“十二五”计算机类规划教材

C++程序设计与应用实验指导及习题解答

◎ 周仲宁　主　编
◎ 叶燕文　副主编



机械工业出版社
CHINA MACHINE PRESS

普通高等教育“十二五”计算机类规划教材

C++程序设计与应用实验指导及习题解答

周仲宁 主 编

叶燕文 副主编

马 彦 曹晓军 彭会萍 参 编

杨海军 鹿 民

机械工业出版社

前　言

C++作为面向对象的程序设计语言，已经成为程序员最广泛使用的工具。很多学校都将C++作为大学生的计算机编程入门语言。本书作者在长期教学实践过程中深切感受到，学生通过看书和听课，对老师所讲和书上所写的内容基本上能够理解，但是当需要自己编程时，却又无从下手。对于学生或自学者来说，要想提高编程能力，需要不断地实践、再实践。然而，很多时候学生会因为花费四、五个小时调不通一个简单程序而失去学习的兴趣。

本书作为《C++程序设计与应用（7-111-25686-1）》配套的学习用书，就是为了读者能在实践过程中随时随地得到指导，起到提高读者编程能力的作用。本书的章节安排与《C++程序设计与应用》相对应，既可以配合该书使用，也可以独立使用。本书内容的选取与定夺是作者多年教学与科研中使用C++经验的总结积累。

应该说明，本书给出的程序并非是唯一正确的解答，我们只是提供了一种参考方案，读者完全可以编写出更好的程序。本书所有的程序都在Visual C++6.0环境下调试通过。

本书由周仲宁任主编，叶燕文任副主编，马彦、曹晓军、彭会萍、杨春林、鹿民参编。其中，叶燕文编写第5、6、8、12、20章；马彦编写第9、10、18、19章；曹晓军编写第1、11、15、17章；彭会萍编写第2、3、13、14章；杨海军编写第4、7章；鹿民编写第16章。

书中可能存在不少的错误与不足，敬请阅读本书的老师和同学们予以批评指正。

编　者

目　录

第 1 章　C++语言概述

1.1　学习指导

1.1.1　基本要求

1. 了解 C++的特点。
2. 了解计算机语言的发展过程。
3. 了解结构化程序设计和面向对象程序设计的基本思想。
4. 掌握程序开发的步骤。
5. 了解一个 C++程序的各个组成部分。

1.1.2　核心内容

1. C++的特点

C++完全兼容 C，但比 C 更可靠，许多 C 代码不经修改就可为 C++使用。

用 C++编写的程序可读性好，代码结构合理，可直接在程序中映射问题空间的结构。

C++生成的代码质量高，软件在可重用性、可扩充性、可维护性和可靠性等方面有更大提高，使大中型程序的开发变得更容易。

C++支持面向对象的机制，可方便地构造出模拟现实问题的实体和操作。

2. 结构化程序设计的特点

结构化程序设计主张使用顺序、选择、循环 3 种基本结构来解决任意复杂的问题。其要点是“自顶而下，逐步求精”，即从问题的总体目标开始，抽象低层的细节，先专心构造高层的结构，然后再一层一层地分解和细化。程序编写时，所有模块的功能通过相应的子程序（函数或过程）的代码来实现。程序的主体是子程序层次库，它与功能模块的抽象层次相对应，模块划分时应追求低耦合和高内聚，模块的编码原则应使程序流程简洁、清晰，可读性强。

3. 面向对象程序设计（OOP）的 3 个重要特性

（1）封装性　对象是数据和处理该数据的方法所构成的整体，外界只能看到其外部特性（消息模式、处理能力等），其内部特性（私有数据、处理方法等）对外不可见。对象的封装性使得信息具有隐蔽性，它减少了程序成分间的相互依赖，降低了程序的复杂性，提高了程序的可靠性和数据的安全性。

（2）继承性　继承性反映的是类与类之间的不同抽象级别，根据继承与被继承的关系，可分为基类和派生类。继承性使得相似的对象可以共享程序代码和数据，继承性是实现程序可重用性的关键。

（3）多态性　多态性是指在形式上表现一样的方法，根据传递给它的参数的不同，可以调用不同的方法体，实现不同的操作。

4. 使用 C++开发一个应用程序的步骤

1）首先要根据实际问题确定编程思路，包括程序设计方法的选择以及解决问题的模型的选择。

2）根据编程思路或解决问题的模型编写程序。

3）编辑源程序。

4）编译和连接。

5）上机反复调试程序，直到改正了所有的编译错误和运行错误。

6）运行。

1.2 实验

实验题目：C++开发环境

1.2.1 实验目的

1. 了解和使用 Visual C++集成开发环境（简称为 IDE）。

2. 熟悉 Visual C++ 6.0 环境的基本命令和功能键，熟悉常用的功能菜单命令。

3. 学习使用 Visual C++ 6.0 环境的帮助。

4. 学习完整的 Visual C++ 6.0 程序开发过程。

5. Visual C++ 6.0 工程相关文件介绍。

6. Visual C++ 6.0 的错误信息。

7. Visual C++ 6.0 的调试工具及使用方法，包括程序断点的设置与去除，程序的跟踪以及运行期间变量数值的观察。

1.2.2 实验内容和步骤

1. Visual C++集成开发环境介绍

Visual C++是微软 Visual Studio（又称 Developer Studio）程序设计软件包的组成之一。Visual Studio 是一个通用的应用程序集成开发环境，包含了一个文本编辑器、资源编辑器、工程编译工具、一个增量连接器、源代码浏览器、集成调试工具以及一套联机文档。使用 Visual Studio，可以完成创建、调试、修改应用程序等各种操作。

2. Visual C++的安装和启动

Visual C++是 Visual Studio 的一部分，因此需要用 Visual Studio 光盘，执行其中的 setup. exe，并按屏幕上的提示安装即可。

在需要使用 Visual C++时，只需从桌面上顺序选择“开始—程序—Microsoft visual Studio—Visual C++ 6.0”命令即可，此时屏幕上在短暂显示 Visual C++ 6.0 的版权页面后，出现 Visual C++ 6.0 的由窗口、工具条、菜单、工具及其他部分组成的一个主界面。通过这个界面，用户可以创建、测试、调试应用程序。如图 1-1 所示。

3. 常用功能键及其意义

为了使程序员能够方便快捷地完成程序开发，开发环境提供了大量快捷方式来简化一些常用操作的步骤。键盘操作直接、简单，而且非常方便。表 1-1 是一些最常用的功能键。

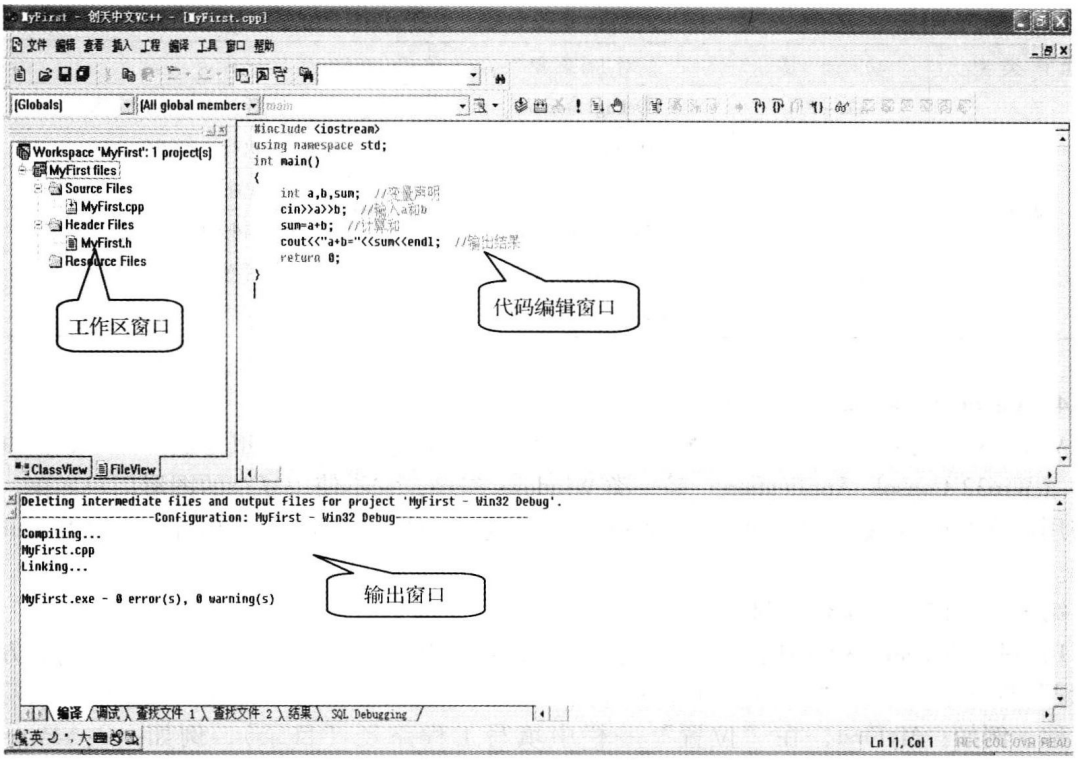


图 1-1 Visual C++ 6.0 界面

表 1-1 常用功能键

操作类型	功能键	对应菜单	含义
文件操作	Ctrl + N	File|New	创建新的文件、项目等
	Ctrl + O	File|Open	打开项目、文件等
	Ctrl + S	File|Save	保存当前文件
编辑操作	Ctrl + X	Edit|Cut	剪切
	Ctrl + C	Edit|Copy	复制
	Ctrl + V	Edit|Paste	粘贴
	Ctrl + Z	Edit|Undo	撤销上一个操作
	Ctrl + Y	Edit|Redo	重复上一个操作
	Ctrl + A	Edit|SelectAll	全选
	Del	Edit|Del	删除光标后面的一个字符
建立程序操作	Ctrl + F7	Build|Compiler current file	编译当前源文件
	Ctrl + F5	Build|Run exe	运行当前项目
	F7	Build|Build exe	建立可执行程序
	F5	Build|Start Debugging	启动调试程序
调试	F5	Debug|Go	继续运行
	F11	Debug|Step into	进入函数体内部

（续）

操作类型	功 能 键	对应菜单	含 义
调试	shift + F11	Debug| Step out	从函数体内部运行出来
	F10	Debug| Step over	执行一行语句
	F9		设置/清除断点
	Ctrl + F10	Debug| Run to cursor	运行到光标所在位置
	shift + F9	Debug| QuickWatch	快速查看变量或表达式的值
	Shift + F5	Debug| Stop debugging	停止调试

4. Visual C++ 6.0 程序开发过程

Visual C++提供了一种控制台操作方式，初学者使用它应该从这里开始。Win32 控制台程序（Win32 Console Application）是一类 Windows 程序，它不使用复杂的图形用户界面，程序与用户交互时通过一个标准的正文窗口，通过几个标准的输入输出流（I/O Streams）进行。

第 1 步：建立控制台工程。

1）进入 Visual C++环境后，选择菜单“文件|新建”，在弹出的对话框中单击上方的“工程”选项卡，选择“Win32 Console Application”工程类型，在“工程”文本框中填写工程名称，例如：MyFirst，在“位置”一栏中填写工程路径（目录），例如：C:\Program Files\Microsoft Visual Studio \ MyProjects \ MyFirst，如图 1-2 所示，然后按“确定”按钮继续。

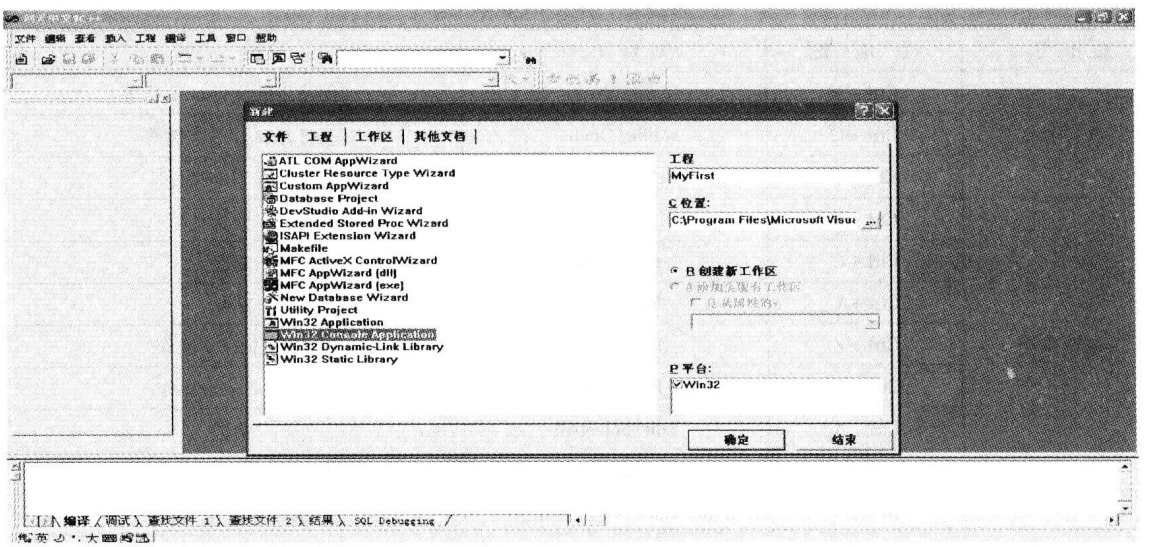


图 1-2 创建控制台工程

2）在“Win32 Console Application—Step 1 of 1”对话框中，选择“An empty project”项，然后按“Finish”按钮继续。

3）在“新建工程信息”对话框中，按“确定”按钮完成工程创建。

第 2 步：编辑 C++程序。

1）选择菜单“添加到工程|文件”，为工程添加新的 C++源文件。如图 1-3 所示。

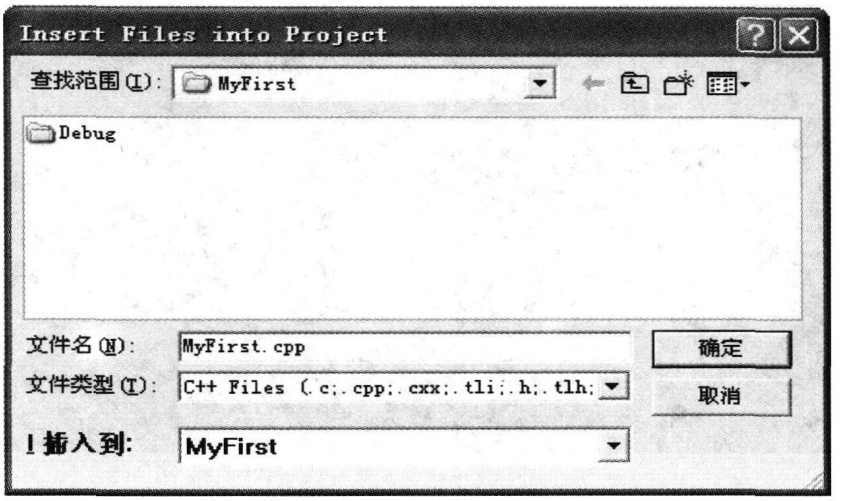


图 1-3　加入新的 C++Source File

2）打开 MyFirst. cpp 的编辑窗口，编辑源文件。代码如下：

```
#include < iostream >
using namespace std;
int main( )
{
int a,b,sum;  //变量声明
cin >> a >> b;  //输入 a 和 b
sum = a + b;  //计算和
cout << "a + b = " << sum << endl;  //输出结果
return 0;
}
```

第 3 步：编译源程序。

选择“编译|编译 MyFirst. cpp”，系统将会在“输出”窗口给出所有的错误信息和警告信息。当所有错误修正之后，系统生成扩展名为 . exe 的可执行文件。对于“输出”窗口给出的错误信息，双击可以使输入焦点跳转到引起错误的源代码处以进行修改。

第 4 步：执行程序。

选择“编译| 执行 MyFirst. exe”菜单项，执行程序，将会出现一个 DOS 窗口，按照程序输入要求正确输入数据后，程序即正确执行。如图 1-4 所示。

第 5 步：调试程序。

编写的程序若已没有编译错误，可以成功运行，但能够运行仅仅意味着程序中不存在语法错误，并不能够说明程序中没有逻辑错误，逻辑错误会导致程序运行结果不正确，这就需要使用 VC 提供的程序调试手段。

在工具栏上单击鼠标右键，在弹出的菜单中选中“调试”项。在程序调试状态下，可以

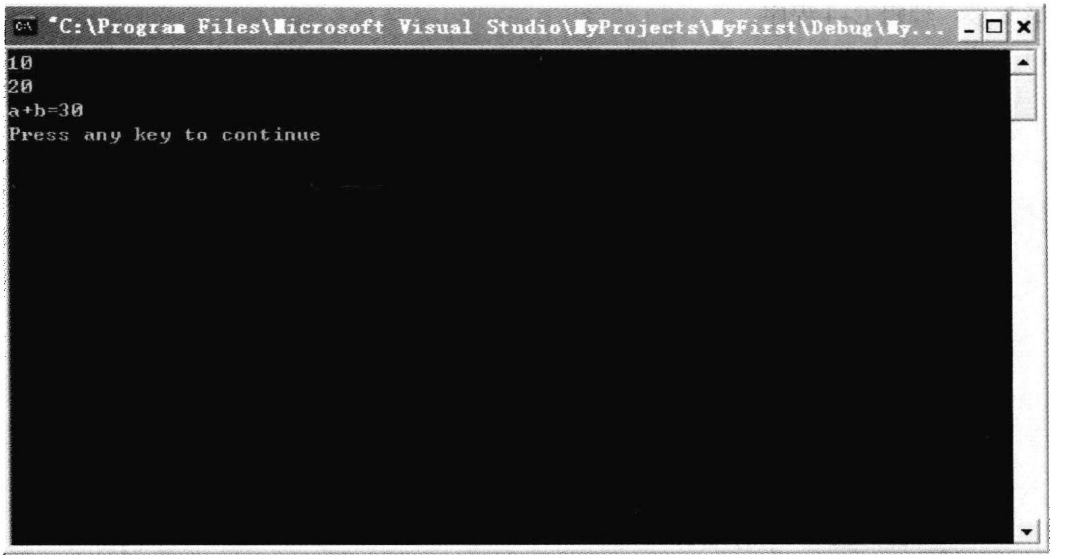


图 1-4　程序运行结果

进行单步执行调试程序。其中，单步跟踪进入子函数（Step Into，F11 快捷键），每按一次 <F11> 键，程序执行一条无法再进行分解的程序行；单步跟踪跳过子函数（Step Over，F10 快捷键），每按一次 <F10> 键，程序执行一行；Watch 窗口可以显示变量名及其当前值，在单步执行的过程中，可以在 Watch 窗口中加入所需观察的变量，辅助加以进行监视，随时了解变量当前的情况；同时，为方便较大规模程序的跟踪，可以设置断点（F9 为快捷键），断点处所在的程序行的左侧会出现一个红色圆点。选择“Build | Start Debug | Go”时，程序执行到断点处程序将暂停执行，可方便用户进行变量观察。取消断点只需在代码处再次按 <F9> 键即可。

5. Visual C++工程相关文件

Visual C++工程相关文件见表 1-2。

表 1-2　Visual C++工程相关文件

序　号	文 件 名	描　述
1	*. dsw	VC 工作区文件
2	*. dsp	（Developer Studio Project）项目文件，文本格式。不熟悉的话不要手工修改
3	*. ncb	无编译浏览文件。当自动完成功能出问题时，可以删除此文件，编译工程后会自动生成
4	*. opt	工程关于开发环境的参数文件，如 VC 工具条位置信息等
5	*. h	C/C++程序头文件，可用文本编辑器打开
6	*. cpp	C++源程序文件，可用文本编辑器打开
7	*. exe	程序可执行文件
8	*. plg	编译信息文件
9	*. pch	（Pre-Compiled File）是预编译文件。可以加快编译速度，但是文件非常大
10	*. pdb	（Program Database）记录了程序有关的一些数据和调试信息，在调试时有用
11	*. obj	源程序文件的目标文件，编译后生成
12	*. ilk	源程序文件的连接文件，连接后生成

注：在 VC 下建立一个工程后，会自动产生 1 ~ 4 类型的文件。加入 *. h、*. cpp 文件编译后产生 7 ~ 12 类型的文件。其中 8 ~ 12 类型的文件位于工程目录下的 Debug 目录下。1 ~ 4 类型的文件都是工程相关的文件，一般不要删除，因为删除后，必须手动地重新建立工程。对于大型项目来说，删除这些文件的后果很严重。5 ~ 6 类型的文件不仅不能删除，还要备份这些文件。失去这些文件是灾难性的。7 ~ 12 类型的所有文件都是可以删除的，因为再次编译连接运行程序时，会再次自动生成这些文件。

6. Visual C++错误信息

程序错误的类型主要有下面 5 种：

1）严重错误（Fatal Error）：很少出现，通常是内部编译器出错。造成编译立即停止。语法错误（Error）是指源程序中存在不符合 C/C++语言语法规定的语句，例如将 int 写成 Int、括号不匹配等。这些错误不改正是不能通过编译的。

2）警告错误（Warning）：对于一些在语法上有轻微错误但不影响程序运行的错误（如定义了变量但始终未使用），编译时会发出警告信息，虽然程序能通过编译、连接、运行，但警告类的错误常常带来程序非法操作、运行错误等问题。所以，应尽量改正警告错误。

3）连接错误（Link Error）：程序语法上没有问题，但是在连接时出现错误。这类问题常常是因为程序与依赖的函数、库不匹配造成的。

4）逻辑错误：逻辑错误是指程序无语法错误，也能正常运行，但结果不对。这类错误常常是设计算法时的错误，计算机无法检查出来。逻辑错误是最难改正的错误之一，引起错误的原因往往可能很不起眼，比如只是一个变量没有初始化等，所以改正这类错误常常需要投入大量的精力。

5）运行错误：有时程序即使无语法和逻辑错误，但是程序仍不能正常运行。多数情况下是输入数据和程序要求的数据不匹配造成的，也可能是系统的支持问题。

在 C++中，语法、连接错误相对较为容易改正。而逻辑错误是最隐蔽的错误，比较难以改正。运行错误则主要是在程序强壮性、兼容性上可能存在问题，可以通过提高程序的适应能力来修正。而最容易让程序开发人员忽略的就是警告错误了，因为警告错误不一定会影响程序的运行，但正是这种不确定性也传递给了程序的执行，不知道程序什么时候会出问题，也许永远不会有问题，也许问题马上出现，也许问题数年后才出现，所以，一定要重视警告错误。表 1-3 是 C++错误的前缀描述，在遇到错误时根据其前缀就知道错误等级了。

表 1-3 C++错误的前缀描述

错误等级	错误前缀	错误编号范围	示 例
严重错误	C1	001 ~ 999	C1001：内部编译器错误（编译器文件 file，行号），编译器无法生成正确的构造代码，原因可能是出自表达式与优化选项的组合
编译错误	C2	001 ~ 999	C2065：The specified identifier was not declared.（标识符，比如一个变量，未申明）
警告错误	C4	001 ~ 999	C4101：'main'：unreferenced local variable（局部变量申明后从未使用过）
连接错误	LNK	1000 ~ 6026	LNK2001：unresolved external symbol "symbol".（连接指定的函数或库异常）

7. 错误信息的查询

只有详细地了解了错误的原因，才能正确地改正错误。Microsoft 的 MSDN Library（Microsoft 开发帮助文档库）提供了查询错误信息的功能。具体使用方法如下：

1）选择 MSDN Library（见图 1-5）的索引属性页，在“键入要查找的关键字：”中输入错误号（由错误前缀和编号组成），比如输入 C1001。

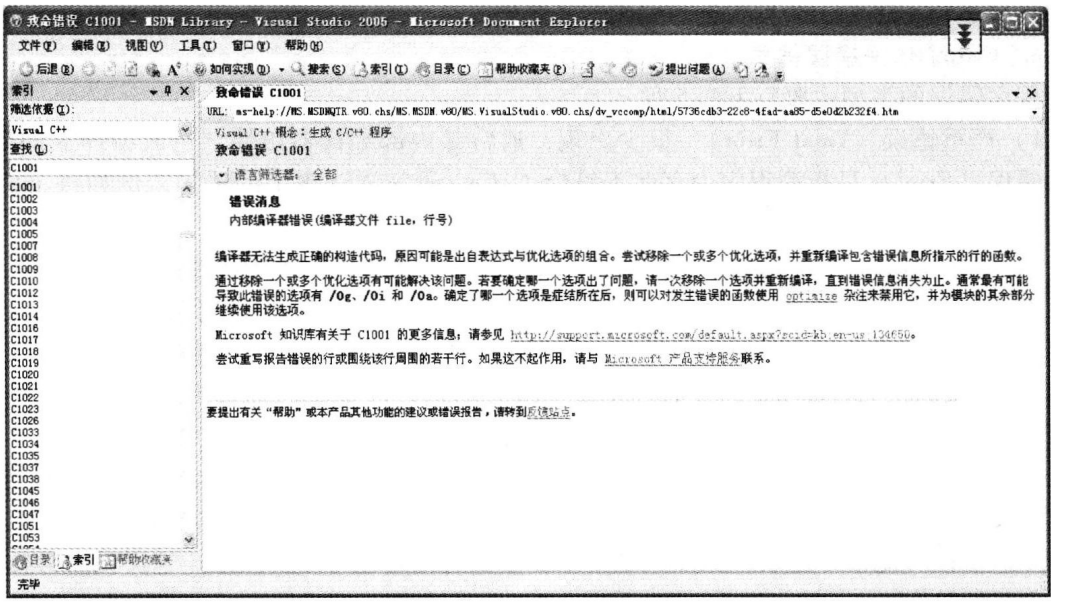


图 1-5　MSDN Library

2）在索引列表框中双击选择匹配的错误后，错误类型、可能原因、改正方法等信息则显示在窗口的右边。

1.3　习题解答

1. C++语言有哪些主要特点？C++程序的结构是怎样的？

解：C++的特点主要表现为 4 个方面。第一，C++完全包含 C，但是对 C 的类型系统进行了改革和扩充，比 C 更可靠，是更好的 C。第二，用 C++编写的程序可读性好，代码结构合理，可直接在程序中映射问题空间的结构。第三，C++生成的代码质量高，软件在可重用性、可扩充性、可维护性和可靠性等方面有更大提高，使大中型程序的开发变得更容易。第四，C++支持面向对象的机制，可方便地构造出模拟现实问题的实体和操作。

2. 简述计算机程序设计语言的发展阶段。

解：程序设计语言的发展共经历了五代。

第一代程序设计语言（1GL）是机器语言，直接使用 0 和 1 的序列作为机器指令来编程，不需要编译和转换就能够被 CPU 直接使用。

第二代程序设计语言（2GL），通常是指一些形式的汇编语言。

第三代程序设计语言（3GL），即高级语言。如 FORTRAN 语言、COBOL、BASIC 语言、C 语言。

第四代语言（4GL）是比较典型的高级语言，它是更接近于人类语言的程序设计语言。是一种非过程化的语言，它提高了软件发展的过程。

第五代程序设计语言（5GL）是使用赋予这个语言的约束而不是使用由一个程序员写的算法、立足于解决问题的一种程序设计语言。第五代语言主要应用在人工智能研究上，Prolog、OPS5 和 Mercury 是最知名的第五代语言。

3. 简述结构化和面向对象程序设计的基本思想和它们之间的联系与区别。

解：结构化程序设计方法主张使用顺序、选择、循环 3 种基本结构来嵌套连接成具有复杂层次的“结构化程序”，严格控制 GOTO 语句的使用。结构化程序设计的要点是：“自顶而下，逐步求精”的设计思想，“独立功能，单入口、单出口”的模块仅用 3 种基本控制结构的编码原则。

面向对象方法的基石是对象和类，其基本机制是方法和消息。OOP 有 3 个重要特性：封装性、继承性和多态性。OOP 方法是以“对象”为中心进行分析和设计，其解决过程从总体上说是采用自底向上方法，先将问题空间划分为一系列对象的集合，再将对象集合进行分类抽象，一些具有相同属性行为的对象被抽象为一个类，类还可抽象分为子类、超类（超类是子类的抽象）。它有许多优点：以“对象”为中心的设计方式能够再现人类认识事物和解决问题的方式；OOP 方法以对象为唯一的语义模型，整个软件任务是通过各对象（类）之间相互传递消息的手段协同完成，能尽量逼真地模拟客观世界及其事物；由于对象和类实现了模块化，类继承实现了抽象对象，以及对象的内部状态和功能实现的细节对外都是不可见的，因此很好地实现了信息隐藏。建立在类及其继承性基础上的重用能力可应付复杂的大型的软件开发。面向对象方法使得软件具有良好的体系结构，便于软件构件化和软件复用，使软件具有良好的扩展性和维护性，抽象程度高，因而具有较高的生产效率。

4. 简述程序开发的步骤和运行过程。

解：1）首先要根据实际问题确定编程思路，包括程序设计方法的选择以及解决问题的模型的选择。

2）根据编程思路或解决问题的模型编写程序。

3）编辑源程序。

4）编译和连接。

5）上机反复调试程序，直到改正了所有的编译错误和运行错误。

6）运行。

第 2 章　数据类型与表达式

2.1　学习指导

2.1.1　基本要求

1. 了解 C++的标识符及规定。

2. 掌握 C++的基本数据类型，了解每种类型在不同的环境下所占的字节数和所能表示的数值范围。

3. 了解 C++的枚举、结构、联合等自定义数据类型的定义方法和简单应用。

4. 了解常量和变量的区别，掌握变量中对每种类型的不同规定。

5. 掌握几种运算符的表示形式和运算符的优先级、结合方向。

2.1.2　核心内容

1. C++的数据类型及其分类

C++的数据类型分为基本类型和自定义类型。

基本类型对应于计算机的基本存储单元和使用这些单元保存数据的一些最常用的操作，如：布尔类型（bool）、字符类型（char、wchar_t）、整型（int）、浮点型（float、double）等。因为字符类型从本质上来说也是整数类型，它是长度为 1 个字节的整数，通常用来存放 ASCII 码，因此将布尔型、字符型和整数类型统称为整型（Integral Type），而将整型和浮点型一起称为算术类型。

自定义类型包括表示一组特定值的枚举类型（enum）、指针类型（如 int * ）、数组类型（如 char[ ]）、引用类型（如 double&）、结构（struct）和类（class）等。

不同数据类型运算时，存在类型转换（隐式自动转换和强制转换）。

2. 常量和变量

C++程序所处理的数据，除有数据类型的区分之外，还有常量和变量之分。

所谓常量是指在程序运行的整个过程中其值始终不变的数据。C++程序的常量可以分为字面常量（Literal Constant）和符号常量，字面常量用符号直接表示，而符号常量则需要使用 const 关键字定义。符号常量在使用之前一定要首先定义，这一点与变量很相似。常量定义的语法格式如下：

const 类型说明符 常量名 = 值;

或者

类型说明符 const 常量名 = 值;

变量是指在程序的执行过程中其值可以变化的量。变量是用来存储数据的值的空间，变量需要用名字唯一标识，像常量的类型一样，变量也具有相应的类型。程序中，变量定义的作用

是指定变量的具体类型和名字。定义变量时，除了指定变量类型和名字之外，还可以指定变量的存储类型，如静态变量、寄存器变量和自动存储变量等。定义变量的语法格式如下：

类型说明符 变量名表；

3. 表达式及表达式语句

表达式是由运算符和操作数构成的可计算的式子，其中运算符是表示进行某种运算的符号，操作数可以是常量、变量、函数调用等，甚至可以是另一个表达式。

由表达式构成的语句称为表达式语句，其作用是要求计算机按照表达式来求得计算结果，并且执行某些操作。

4. 几种运算符及优先级

（1）算术运算符

算术运算符包括 +（正号）、-（负号）、+（加）、-（减）、*（乘）、/（除）、%（取余）。其中，+（正号）、-（负号）的优先级最高，接下来是*（乘）、/（除），然后是%（取余），优先级最低的是 +（加）、-（减）。学习过程中应注意整数除法规则的不同及取余运算只在整型量之间进行。

（2）自增自减运算符

自增（++）、自减（--）运算符有两种应用格式：

运算符后置用法，代表先使用变量，再对变量自增（自减）。

运算符前置用法，代表先对变量自增（自减），再使用变量。

自增与自减运算符只能和左值表达式一起使用。前置自增（自减）表达式具有左值特征，而后置自增（自减）表达式不具有左值特征。

（3）赋值运算符

赋值运算符表示把一个表达式的结果值赋给一个变量。C++提供了最简单的赋值运算符“=”及其复合赋值运算符（+=、-=、*=、/=、%=等）。

（4）逗号运算符

在 C++中，多个表达式可以用逗号分开。其中用逗号分开的表达式被从左至右分别计算，并且整个表达式的值是最后一个表达式的值。

（5）关系与逻辑运算符

关系运算符用于两个表达式的值进行比较，返回值为 true（真）或 false（假），分别用值 1（true）或 0（false）表示。

关系运算符都是双目运算符，其结合性是从左到右，其中，关系运算符中 <、<=、>、>= 运算符的优先级相同，== 和 != 运算符的优先级相同。前者运算符的优先级高于后者。

逻辑运算符根据表达式的真/假属性返回真值或假值。包括！（逻辑非）、&&（逻辑与）、||（逻辑或）。其中逻辑非的优先级最高，逻辑与次之，逻辑或最低。

（6）条件运算符

条件运算符“?:”是 C++中唯一的三目运算符，是对第一个表达式的真/假结果进行检验，然后根据检验结果将之后两个表达式中的一个作为整个表达式的结果。

（7）sizeof 运算符

sizeof 运算符用于计算某种类型的对象在内存中所占的字节数。