



普通高等教育信息技术类系列规划教材

C语言程序设计实验指导

唐新来 王 萌 主编



科学出版社
www.sciencep.com

普通高等教育信息技术类系列规划教材

C 语言程序设计实验指导

唐新来 王 萌 主 编

李春贵 甘秋玲 彭 颖 副主编

科学出版社

北 京

内 容 简 介

本书是《C 语言程序设计》的配套教材, 共分为 4 章。第 1 章为 Visual C++ 集成开发环境, 介绍了 C 语言上机环境及其基本操作; 第 2 章为实验, 介绍了 10 个实验及常见的上机错误信息; 第 3 章为单元测试, 根据教学要求, 精选了相应章节的习题, 主要有选择题和填空题两种题型; 第 4 章为等级考试, 介绍了计算机等级考试(二级)的相关知识点并提供了模拟试题。

本书通俗易懂、深入浅出、实用性强, 可作为独立学院(三本)、高职高专院校计算机相关专业的《C 语言程序设计》课程实验教学用书, 同时可供普通高等院校非计算机专业的学生、计算机培训班的学员、参加计算机等级考试人员使用, 还可作为 C 语言程序设计自学者的参考书。

图书在版编目(CIP)数据

C 语言程序设计实验指导/唐新来, 王萌主编. —北京: 科学出版社, 2009
(普通高等教育信息技术类系列规划教材)

ISBN 978-7-03-025445-0

I. C… II. ①唐…②王… III. C 语言—程序设计—高等教育—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 154313 号

责任编辑: 李太铕 文 戈 / 责任校对: 耿 耘

责任印制: 吕春珉 / 封面设计: 一克米工作室

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

骏主印刷厂印刷

科学出版社发行 各地新华书店经销

*

2009 年 9 月 第 一 版 开本: 787×1092 1/16

2009 年 9 月 第一次印刷 印张: 10 1/4

印数: 1-3 000 字数: 234 000

定价: 18.00 元

(如有印装质量问题, 我社负责调换<环伟>)

销售部电话 010-62134988 编辑部电话 010-62135763-8220

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

前 言

本书是《C 语言程序设计》的配套教材。全书共分为 4 章，第 1 章介绍了 C 语言上机环境及其基本操作；第 2 章介绍了 10 个实验及常见的上机错误信息；第 3 章根据教学要求，精选了相应章节的习题；第 4 章介绍了计算机等级考试（二级）的相关知识点并提供了模拟试题。

本书的主要内容是 C 语言程序设计实践教学的基础实验。基础实验紧密结合课堂教学，旨在帮助学生加深和巩固重点概念的理解，掌握编程和程序调试的基本方法，提高编程和程序调试能力。基础实验题目分测试型、模仿型和设计型三类，以设计型为主。基础实验一般规模较小，难度不大，初学者容易掌握。

本书的另一个重要内容是提供了精选习题和试题。习题和试题选用了以下四种题型，每类题型有各自的侧重点。

- (1) 阅读程序：重点是概念的应用和阅读程序的基本方法。
- (2) 程序改错：重点是概念的应用及发现和排除程序错误的方法。
- (3) 程序填空：重点是问题分析和基础算法的掌握。
- (4) 编写程序：重点是问题分析、算法设计和编程技巧的综合应用。

另外，考虑到学生参加相应计算机等级考试的要求，本书给出了等级考试的考试大纲、模拟试题及参考答案，帮助读者了解计算机等级考试的特点从而更顺利地通过考试。

全书在内容上侧重于重点概念的理解和应用以及问题分析和问题求解方法的探讨，是课外复习和自学的参考书。

读者在课外复习和自学时，应自己先分析、解答问题，然后再参考书中答案。这样能加深对概念的理解，开阔思路，掌握方法，提高能力。书中程序全部在 Visual C++ 6.0 环境下调试并运行通过。

本书由唐新来、王萌任主编，李春贵、甘秋玲、彭颖任副主编。全书由广西工学院的李春贵、何春华、王萌、王晓荣，广西工学院鹿山学院的唐新来，桂林理工大学博文管理学院的唐杰，广西大学行健文理学院的甘秋玲、彭颖等联合编写完成。其中，第 1 章的 1.1 节到 1.2.5 节由李春贵编写，第 1.2.6 节和 1.2.7 节由唐新来编写，第 2.1 节由唐杰编写，第 2.2 节由王晓荣编写，第 2.3 节由甘秋玲、彭颖编写，第 3 章由王萌编写，第 4 章由何春华编写。王萌负责本书的统稿。

本书在编写过程中得到了广西工学院和广西工学院鹿山学院的大力支持，并得到有关专家、教师的指导和帮助，在此一并表示衷心的感谢。

由于作者水平有限，书中难免出现疏漏，欢迎专家和读者多提宝贵意见。

编 者

目 录

第 1 章 Visual C++集成开发环境.....	1
1.1 了解 Visual C++可视化集成开发环境.....	1
1.1.1 了解 Visual C++开发环境的实验操作步骤.....	1
1.1.2 Visual C++ 有关联机帮助	5
1.2 Visual C++ 调试入门	6
1.2.1 概述	6
1.2.2 基本断点调试	7
1.2.3 更多的断点调试	8
1.2.4 调试程序——动态查看变量的值.....	9
1.2.5 调试程序——进程控制.....	10
1.2.6 调试程序——调用堆栈（选读）	11
1.2.7 调试程序——其他调试手段（选读）	11
第 2 章 实验.....	12
2.1 实验目的与要求	12
2.1.1 实验目的	12
2.1.2 实验要求	13
2.1.3 实验报告内容	13
2.2 上机实验.....	14
2.2.1 C 程序上机操作初步.....	14
2.2.2 数据类型与表达式	16
2.2.3 选择结构	18
2.2.4 循环结构	21
2.2.5 数组	24
2.2.6 函数	25
2.2.7 指针	27
2.2.8 复杂数据类型	29
2.2.9 文件	31
2.2.10 预编译和位运算	33
2.3 常见上机错误信息	34
2.3.1 一般错误	34
2.3.2 警告信息	43
2.3.3 严重错误	45

第 3 章 单元测试	46
3.1 C 语言程序设计初步	46
3.2 C 程序控制结构	52
3.3 数组	66
3.4 函数	73
3.5 指针	85
3.6 复杂数据类型	98
3.7 文件	101
3.8 预编译与位运算	107
第 4 章 等级考试	113
4.1 广西区二级 C 语言等级考试	113
4.1.1 考试大纲	113
4.1.2 笔试模拟试题	115
4.1.3 参考答案	130
4.2 全国计算机二级 C 语言等级考试	131
4.2.1 考试大纲	131
4.2.2 笔试模拟试题	134
4.2.3 参考答案	153
4.2.4 上机考试模拟试题	155
4.2.5 参考答案	157
参考文献	158

第 1 章 Visual C++集成开发环境

1.1 了解 Visual C++可视化集成开发环境

1.1.1 了解 Visual C++开发环境的实验操作步骤

微软公司推出的 Visual C++是目前使用极为广泛的基于 Windows 平台的可视化集成开发环境。它和 Visual Basic 等其他软件构成了 Visual Studio (又名 Developer Studio) 程序设计软件包。Visual Studio 是一个通用的应用程序集成开发环境, 包含文本编辑器、资源编辑器、工程编译工具、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。

Visual C++分为学习版、专业版和企业版。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标准的正文窗口和标准的输入输出流 (I/O Streams) 进行。下面我们将对使用 Visual C++编写简单的控制台程序作一个初步的介绍。这里的介绍不包含 C++运行环境 (尤其是 Windows 环境) 下进行开发的内容, 有关这方面的内容请参阅相应开发手册。另外, Visual C++包含丰富的函数库和类库, 学生在设计程序时可以参阅相应的开发指南类书籍。

一般来说, 基于 Visual C++开发环境的实验操作步骤如下。

1. 启动 Visual C++

选择菜单“开始|程序|Microsoft Visual Studio 6.0|Microsoft Visual C++ 6.0”即可运行 Visual C++ (也可在 Windows 桌面上建立一个启动命令快捷方式, 以后可双击该快捷方式运行 Visual C++), 见图 1.1。

2. 创建控制台工程

进入 Visual C++ 环境后, 选择菜单“File | New”, 在弹出的对话框中单击上方的选项卡“Projects”, 选择“Win32 Console Application”工程类型, 在“Project name”一栏中填写工程名, 例如“MyFirst”, 在“Location”一栏中填写工程路径 (目录), 例如 D:\MyProject\MyFirst, 见图 1.2, 然后按“OK”按钮继续。

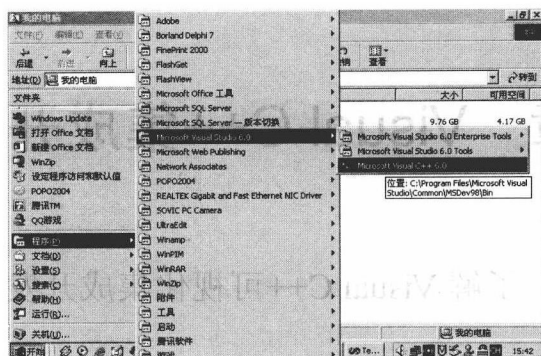


图 1.1 启动 Microsoft Visual C++ 6.0

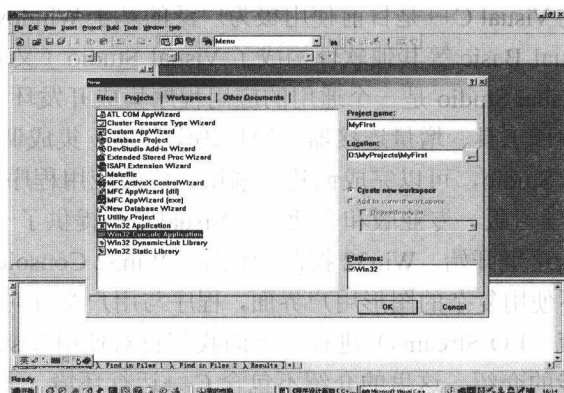


图 1.2 创建控制台工程

接着,屏幕上出现如图 1.3 所示的 Win32 Console Application 对话框,选择“An empty project”项,然后按“Finish”按钮继续,出现如图 1.4 所示的 New Project Information 对话框后,按“OK”按钮完成工程创建。

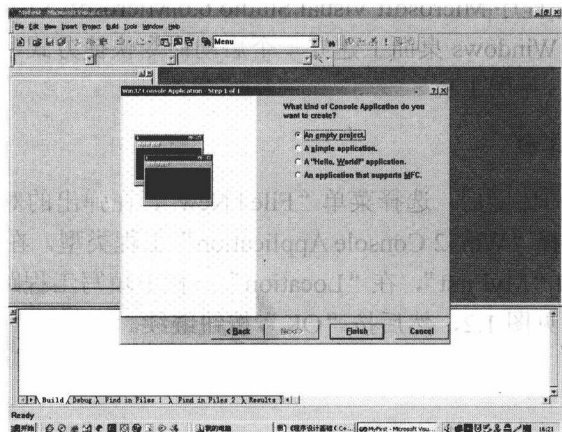


图 1.3 Win32 Console Application 对话框

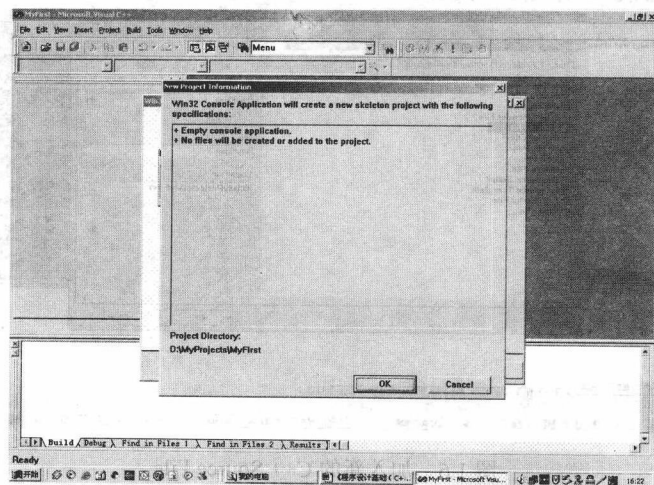


图 1.4 New Project Information 对话框

3. 编辑 C++程序

选择菜单“Project | Add to Project | New”，为工程添加新的 C++源文件，如图 1.5 所示。

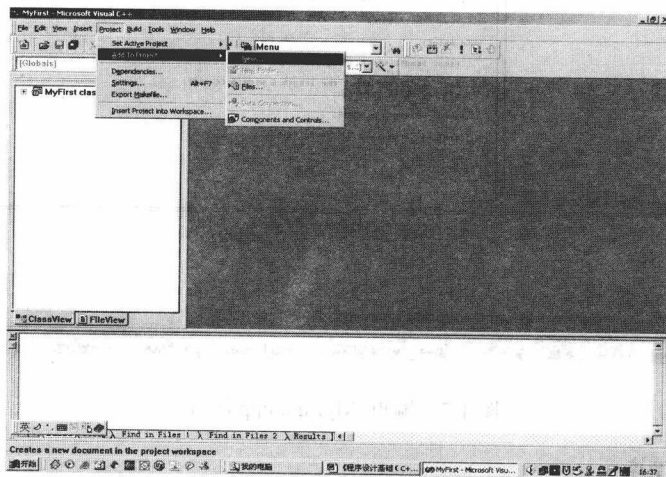


图 1.5 为工程添加新源文件

在出现如图 1.6 所示的对话框后，选择“File”选项卡，选定“C++ Source File”项。在“File Name”栏输入新添加的源文件名，如“MyFirst.cpp”。“Location”一栏指定文件路径，按“OK”按钮完成 C++源程序的系统新建操作，如图 1.6 所示。

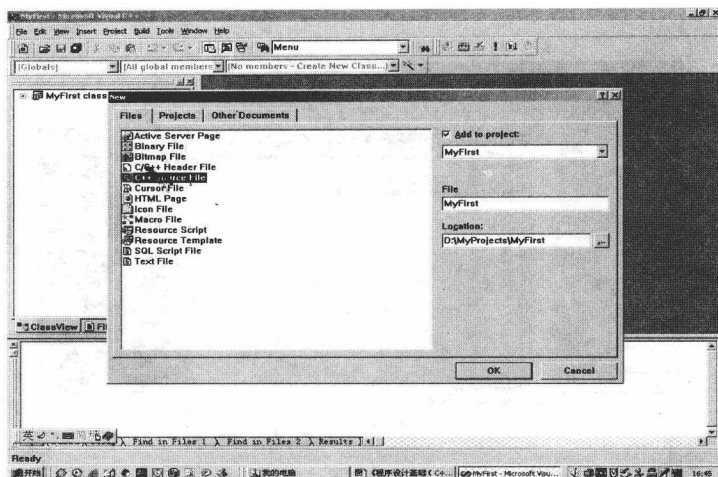


图 1.6 加入新的 C++ Source File

在文件编辑区输入源程序，然后存盘，如图 1.7 所示。

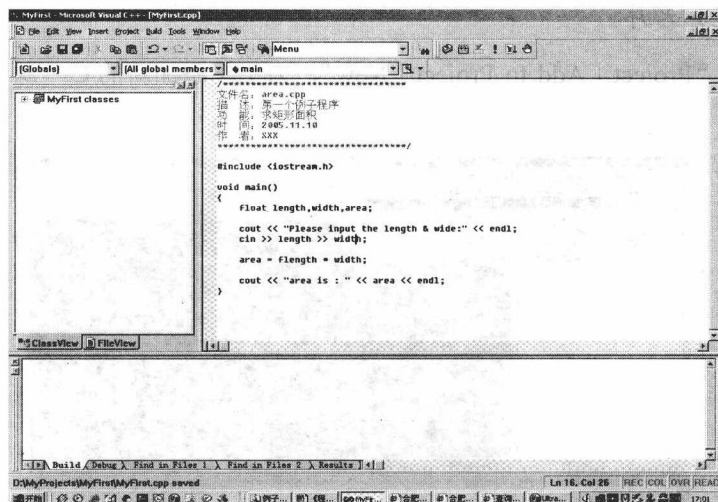


图 1.7 编辑 MyFirst.cpp 源程序

4. 编译源程序

选择“Build | Build”菜单项（F7 为快捷键），系统将会在 Output 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，系统将会生成扩展名为 exe 的可执行文件。对于 Output 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。

5. 执行程序

选择“Build | Execute”菜单项（Ctrl+F5 为快捷键），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 1.8 所示。

6. 调试程序

在编写较长的程序时,能够一次成功而不含有任何错误绝非易事,这需要进行长期大量的练习。编写的程序若已没有编译错误,就可以成功运行。对于程序中的错误,Visual C++提供了易用且有效的调试手段。

在工具栏上单击鼠标右键,在弹出的菜单中选中“Debug”项。在程序调试状态下,可以进行单步执行调试程序。其中,单步跟踪进入子函数(Step Into, F11),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),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 即可。还有运行至当前函数的末尾(Step Out)等都是很有效的调试工具,这里不再做一一介绍,请看 1.2 节。

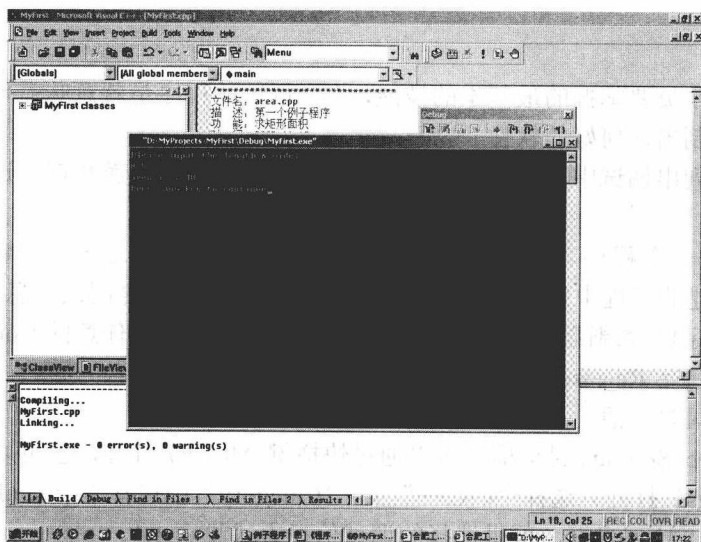


图 1.8 程序运行结果

1.1.2 Visual C++ 有关联机帮助

Visual C++ 6.0 提供了详细的帮助信息,用户通过选择集成开发环境中的“Help”菜单下的“Contents”命令就可以进入帮助系统。在源文件编辑器中把光标定位在一个需要查询的单词处,然后按 F1 键也可以进入 Visual C++ 6.0 的帮助系统,如图 1.9 所示。用户要使用帮助必须安装 MSDN。用户通过 Visual C++ 6.0 的帮助系统可以获得几乎所有的 Visual C++ 6.0 的技术信息,这也是 Visual C++ 作为一个非常友好的开发环境所具

有的一个特色。

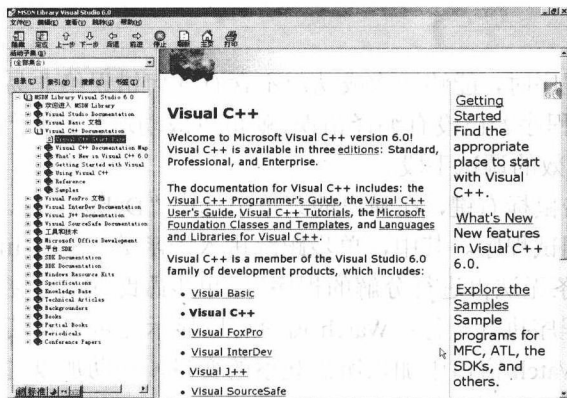


图 1.9 MSDN 联机帮助

1.2 Visual C++ 调试入门

1.2.1 概述

调试是程序员要掌握的最基本的技能，不会调试的程序员就意味着他即使会一门语言，却不能编制出任何好的软件。

这里简要列出调试中比较常用的技术，并且约定，在选择菜单时，通过“|”表示分级菜单设置。

为了调试一个程序，首先必须使程序中包含调试信息。一般情况下，一个从 AppWizard 创建的工程中包含的 Debug Configuration 自动包含调试信息，但是，Debug 版本并不是程序包含调试信息的决定因素，程序设计者可以在任意的 Configuration 中增加调试信息，包括 Release 版本。

为了增加调试信息，可以按照下述步骤进行。

打开 Project Settings 对话框（可以通过快捷键 Alt+F7 打开，也可以通过 IDE 菜单“Project|Settings”打开）选择“C/C++”页，“Category”中选择“general”，则出现一个“Debug Info”下拉列表框，可供选择的调试信息方式见表 1.1。

表 1.1 调试信息可供选择的选项

命令行	Project Settings	说明
无	None	没有调试信息
/Zd	Line Numbers Only	目标文件或者可执行文件中只包含全局和导出符号以及代码行信息，不包含符号调试信息
/Z7	C 7.0- Compatible	目标文件或者可执行文件中包含行号和所有符号调试信息，包括变量名及类型，函数及原型等
/Zi	Program Database	创建一个程序库（PDB），包括类型信息和符号调试信息
/ZI	Program Database for Edit and Continue	除了前面/Zi 的功能外，这个选项允许对代码进行调试过程中的修改和继续执行。这个选项同时使#pragma 设置的优化功能无效

选择“Link”页，选中复选框“Generate Debug Info”，这个选项将使连接器把调试信息写进可执行文件和 DLL。如果“C/C++”页中设置了“Program Database”以上的选项，则“Link incrementally”可以选择。选中这个选项，将使程序可以在上一次编译的基础上被编译（即增量编译），而不必每次从头开始编译。

1.2.2 基本断点调试

断点是调试器设置的一个代码位置，当程序运行到断点时，程序中断执行，回到调试器。调试时，只有设置了断点并使程序回到调试器，才能对程序进行在线调试。设置断点是跟踪程序实际运行流程的最好方法。在 Debug 模式下，程序可以在断点处停止，然后让程序单步执行，来确认程序是否在按照程序设计者所预想的方式运行。

1. 设置断点

可以通过下述方法设置一个断点。

首先把光标移动到需要设置断点的代码行上，然后按 F9 快捷键弹出 Breakpoints 对话框，方法是按快捷键 Ctrl+B 或 Alt+F9，或者通过菜单“Edit|Breakpoints”打开。将鼠标停留在要被暂停的那一行，选择工具栏按钮“Insert|Remove Breakpoint (F9)”，就可以添加一个断点，如图 1.10 所示。

把光标移动到给定断点所在的行，再次按 F9 就可以取消断点。

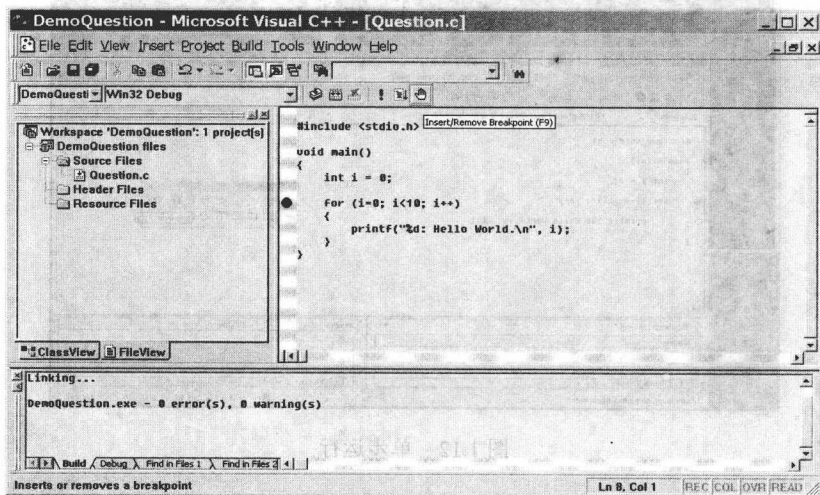


图 1.10 设置断点

2. 开始调试

如图 1.11 所示，选择菜单“Go (F5)”，就可以开始调试程序。

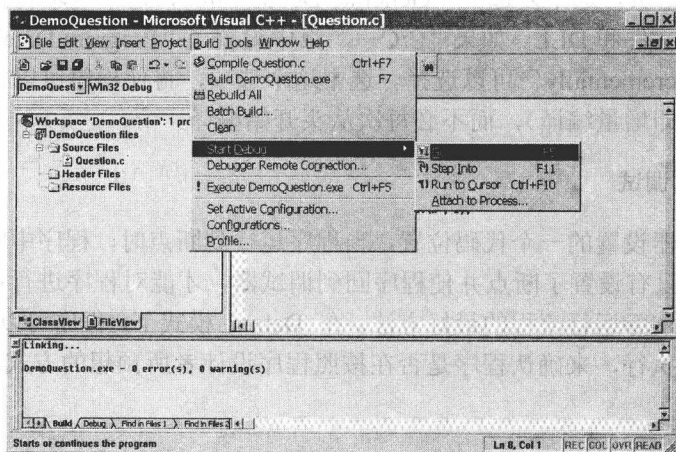


图 1.11 开始调试

3. 单步运行

选择菜单或者工具栏的“Step Over (F10)”按钮，可以进行程序单步运行。不断按 F10 键，程序会一步一步地被执行，如图 1.12 所示。

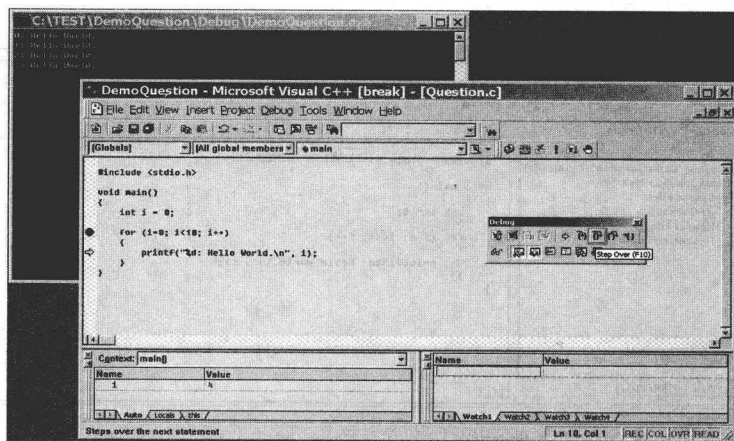


图 1.12 单步运行

1.2.3 更多的断点调试

1. 条件断点

可以为断点设置一个条件，这样的断点称为条件断点。对于新加的断点，可以单击“Conditions”按钮，为断点设置一个表达式。当这个表达式发生改变时，程序就被中断。

2. 数据断点

数据断点只能在 Breakpoints 对话框中设置。选择“Data”页，就显示设置数据断点

的对话框。在编辑框中输入一个表达式，当这个表达式的值发生变化时，就到达数据断点。一般情况下，这个表达式应该由运算符和全局变量构成，例如：在编辑框中输入“g_bFlag”这个全局变量的名字，那么当程序中有“g_bFlag= !g_bFlag”时，程序就将停在这个语句处。

3. 消息断点

Visual C++ 也支持对 Windows 消息进行截获。有两种方式进行截获：窗口消息处理函数和特定消息中断。

在 Breakpoints 对话框中选择“Messages”页，就可以设置消息断点。如果在上面那个对话框中写入消息处理函数的名字，那么每次消息被这个函数处理，就到达断点（如果采用普通断点在这个函数中截获，效果应该一样）。如果在底下的下拉列表框选择一个消息，则每次这种消息到达，程序就中断。

1.2.4 调试程序——动态查看变量的值

Visual C++ 支持查看变量、表达式和内存的值，所有这些观察都必须是在断点中断的情况下进行。Visual C++ 提供一种被称为 Watch 的机制来观看变量和表达式的值。在断点状态下，在变量上单击右键，选择“Quick Watch”，就弹出一个对话框，显示这个变量的值。

单步调试程序的过程中，可以在 Watch 窗口中动态地查看变量的值。单击“Debug”工具条上的“Watch”按钮，就出现一个 Watch 视图（Watch1, Watch2, Watch3, Watch4），在该视图输入变量或者表达式，就可以观察变量或者表达式的值（注意：这个表达式不能有副作用，例如++运算符绝对禁止用于这个表达式中，因为这个运算符将修改变量的值，导致软件的逻辑被破坏）。在 Visual C++ 中，动态查看变量的值有两种方式。

1. 自动查看

观看变量的值最简单，当断点到达时，把光标移动到这个变量上，停留一会通过 Variables（变量）子窗口就可以看到变量的值。

在 Variables 子窗口中会自动显示当前运行上下文中的变量的值，如图 1.13 所示。

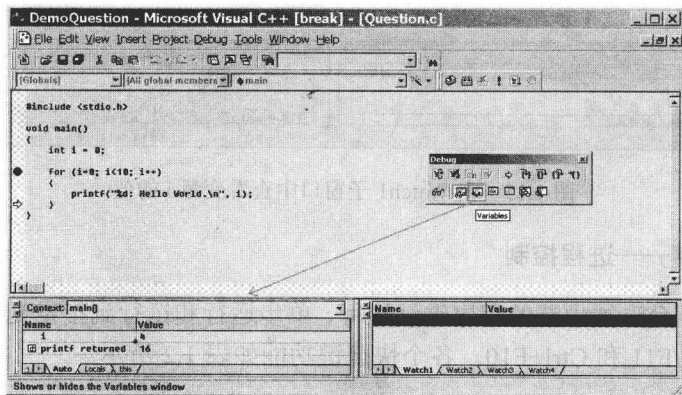


图 1.13 查看上下文中的变量的值

2. 添加变量到 Watch 列表

如果本地变量比较多,自动显示的窗口比较混乱,可以在 Watch 列表中添加自己想要监控的变量名。例如,图 1.14 中在 Watch1 子窗口中添加了变量 i。

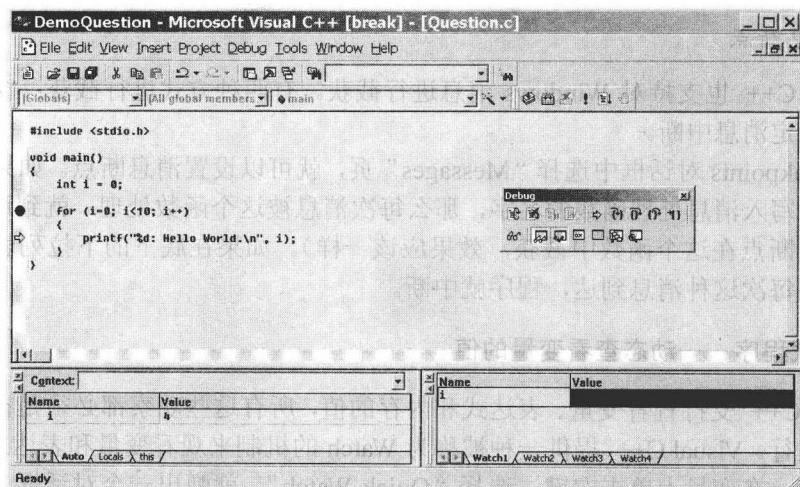


图 1.14 在 Watch1 子窗口中添加了变量 i

添加结束后,该变量的值会被显示出来。并且随着单步调试的进行,会看到变量 i 的值逐渐递增,如图 1.15 所示。

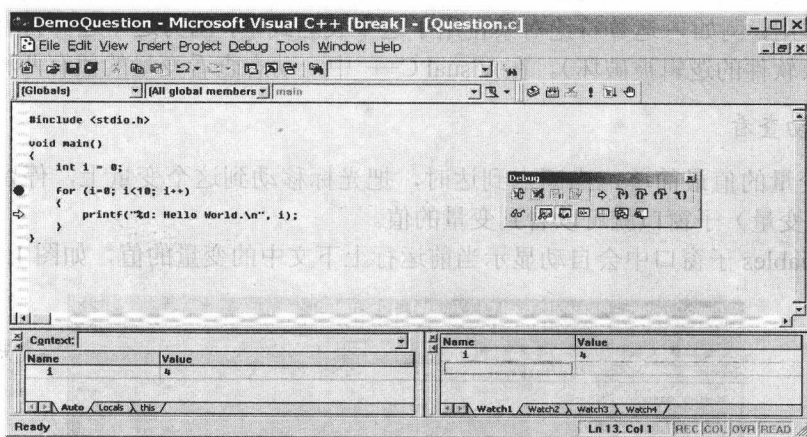


图 1.15 在 Watch1 子窗口中查看变量 i 的值

1.2.5 调试程序——进程控制

Visual C++ 允许被中断的程序继续运行、单步运行和运行到指定光标处,分别对应快捷键 F5、F10/F11 和 Ctrl+F10。各个快捷键功能如表 1.2 所示。

表 1.2 调试快捷键及说明

快捷键	说明
F5	继续运行
F10	单步，如果涉及到子函数，不进入子函数内部
F11	单步，如果涉及到子函数，进入子函数内部
Ctrl+F10	运行到当前光标处

1.2.6 调试程序——调用堆栈（选读）

调用堆栈（Call Stack）反映了当前断点处函数是被哪些函数按照什么顺序调用的。单击“Debug”工具条上的“Call stack”按钮就显示 Call Stack 对话框。在 Call Stack 对话框中显示了一个调用系列，最上面的是当前函数，往下依次是调用函数的上级函数。单击这些函数名可以跳到对应的函数中去。

1.2.7 调试程序——其他调试手段（选读）

Visual C++ 系统提供一系列特殊的函数或者宏来处理 Debug 版本相关的信息，如表 1.3 所示。

表 1.3 调试用函数及说明

宏名/函数名	说明
TRACE	使用方法和 printf 完全一致，它在 output 框中输出调试信息
ASSERT	它接收一个表达式，如果这个表达式为 TRUE，则无动作，否则中断当前程序执行。对于系统中出现这个宏导致的中断，应该认为你的函数调用未能满足系统的调用此函数的前提条件。例如，对于一个还没有创建的窗口调用 SetWindowText 等
VERIFY	和 ASSERT 功能类似，所不同的是，在 Release 版本中，ASSERT 不计算输入的表达式式的值，而 VERIFY 计算表达式式的值