



教育部大学计算机课程改革项目规划教材

C/C++程序设计案例教程 实验与实训指导

梅 险 曲 蔚 孙建业 主编

孙博文 审

高等教育出版社



014032254

TP312C-42
218



教育部大学计算机课程改革项目规划教材

C/C++程序设计案例教程实验 与实训指导

C/C++ Chengxu Sheji Anli Jiaocheng Shiyan
yu Shixun Zhidao

梅险 曲蔚 孙建业 主编
孙博文 审



高等教育出版社·北京



北航

C1720527

TP312C-42

218

内容提要

本书是《C/C++程序设计案例教程——基于计算思维》的配套实验和实践指导教材。本书以培养学生计算思维能力为核心理念，注重工程实践和编程能力的训练；通过实际案例剖析、算法分析和程序代码的设计，培养学生使用C/C++语言编写程序解决实际问题的能力。

本书包括上机实验和综合实践项目两大部分。上机实验部分对应案例教程，涵盖C/C++程序设计语言的数据类型、运算符、表达式、三种控制结构、数组、字符串、指针、函数和结构体类型，以及面向对象程序设计的类与对象、继承与多态、C++输入/输出流类库、异常处理和图形界面编程等内容，共15章；综合实践项目部分由工程和数值计算类、信息处理类、游戏博弈类和其他综合类项目组成，共4章。

本书设计的上机实验项目，不仅注重语法知识点的掌握和应用，更注重对解决问题思路的培养与指导，设计的综合实践项目，注重学生工程性、创新性和研究性能力的培养和训练。通过综合实践，能够有效地帮助学生理解掌握面向对象程序设计方法的核心思想，从而逐步培养学生面向对象的思维方式。

本书适合作为高等学校各专业C/C++语言程序设计课程的实践教材，也可作为参与计算机竞赛和从事计算机相关工作人员的参考资料。

图书在版编目(CIP)数据

C/C++程序设计案例教程实验与实训指导 / 梅险, 曲蔚, 孙建业主编. --北京: 高等教育出版社, 2014.3

ISBN 978-7-04-027085-3

I. ①C… II. ①梅… ②曲… ③孙… III. ①C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第040801号

策划编辑 唐德凯

责任编辑 唐德凯

特约编辑 谷玉春

封面设计 于文燕

版式设计 范晓红

责任校对 殷然

责任印制 张泽业

出版发行 高等教育出版社
社 址 北京市西城区德外大街4号
邮政编码 100120
印 刷 北京丰源印刷厂
开 本 787mm×1092mm 1/16
印 张 15.25
字 数 350千字
购书热线 010-58581118

咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
版 次 2014年3月第1版
印 次 2014年3月第1次印刷
定 价 25.20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

版权所有 侵权必究

物料号 27085-00

前 言

学习程序设计需要大量的上机实验练习和综合项目实践。本书内容主要包括按语法知识点由浅入深循序渐进的实验部分和按应用分类综合运用知识的实践部分，目的是使读者深入理解和掌握 C/C++ 程序设计所涉及的概念、语法、算法与应用技巧。

本书“上机实验”部分共 15 章，主要针对每章实验项目所用的知识进行了总结，又给出了具有代表性的范例。同时与主教材的内容相结合，每章均包含实验目的、知识要点、实验范例和实验习题。实验内容包括程序语言基础、数据和运算、程序结构、数组、字符串、指针、函数、函数重载、用户自定义类型、类和对象、类的继承与多态、输入/输出流、异常处理和图形界面等，涵盖了 C/C++ 程序设计的基本内容与知识点。从配合教学的角度出发，每个实验力求做到覆盖相关章节的所有知识点，以加强对基本理论知识的理解与掌握，锻炼与提高实际编程能力，提高教学效果。

本书“综合实践项目”部分共 4 章，包括工程和数值计算类、信息处理类、游戏博弈类和其他综合类项目，通过完整的题目描述、题目简介、概要设计、详细设计、程序代码和运行界面、分析总结等，从多方面展示程序设计理论在具体项目中的应用，从而培养学生系统分析问题、灵活运用知识、积极发掘创新意识的能力。

本书“附录”部分概要介绍了表示算法流程的符号含义、上机操作常见错误信息及解决方法，同时对学生实验报告和实践报告的撰写提出了明确的内容和格式要求，供使用者参考。本书在习题参考答案部分给出了主教材中部分习题的解题思路和设计方法，希望读者通过学习与实践，能够解决更多的疑难问题。

本书由哈尔滨理工大学计算中心长期从事程序设计教学和科研的一线教师编写。其中，第 1—7 章由曲蔚编写，第 8—14 章由孙建业编写，第 15—19 章和附录由梅险编写，孙博文审阅了全稿。

在本书的编写过程中，得到了哈尔滨理工大学计算中心领导和全体老师的热情帮助，在此表示诚挚的谢意。实践部分题目得到国内著名计算机信息安全研究机构安天实验室和中国人工智能学会机器博弈专业委员会的技术支持，孙德强、张博雅、迟望达、杜世锦等也对实践程序的测试、优化与改进做出了大量的辅助工作，这里一并表示感谢。

由于编者水平有限，书中难免存在疏漏和错误，敬请读者和同行专家指正。

编 者

2013 年 12 月

目 录

第一部分 上机实验

第 1 章 集成开发环境与程序基本框架3	
1.1 实验目的.....3	
1.2 知识要点.....3	
1.3 实验范例.....10	
1.4 实验习题.....11	
第 2 章 数据类型、运算符与表达式14	
2.1 实验目的.....14	
2.2 知识要点.....14	
2.3 实验范例.....15	
2.3.1 数据类型与表达式的应用.....15	
2.3.2 交换变量的值.....16	
2.4 实验习题.....18	
第 3 章 顺序结构程序设计22	
3.1 实验目的.....22	
3.2 知识要点.....22	
3.3 实验范例.....23	
3.3.1 运算符和数学函数的基本应用.....23	
3.3.2 计算周长、面积与体积.....24	
3.4 实验习题.....25	
第 4 章 选择结构程序设计30	
4.1 实验目的.....30	
4.2 知识要点.....30	
4.3 实验范例.....32	
4.3.1 计算分段函数.....32	
4.3.2 成绩等级判定.....33	
4.4 实验习题.....35	
第 5 章 循环结构程序设计40	
5.1 实验目的.....40	
5.2 知识要点.....40	
5.3 实验范例.....41	
5.3.1 奇数数列求和.....41	
5.3.2 阶乘数列求和.....42	
5.4 实验习题.....44	
第 6 章 数组49	
6.1 实验目的.....49	
6.2 知识要点.....49	
6.3 实验范例.....50	
6.3.1 一维数组的应用.....50	
6.3.2 二维数组的应用.....51	
6.4 实验习题.....52	
第 7 章 字符串58	
7.1 实验目的.....58	
7.2 知识要点.....58	
7.3 实验范例.....58	
7.3.1 字符串的基本应用.....58	
7.3.2 字符串的比较与子串删除.....60	
7.4 实验习题.....62	
第 8 章 指针65	
8.1 实验目的.....65	
8.2 知识要点.....65	
8.3 实验范例.....65	
8.3.1 指针的基本应用.....65	
8.3.2 指针的移动.....67	
8.4 实验习题.....68	
第 9 章 函数72	
9.1 实验目的.....72	
9.2 知识要点.....72	
9.3 实验范例.....74	
9.3.1 简单计算器.....74	
9.3.2 数组作为函数参数的应用.....76	
9.4 实验习题.....78	

II 目录

第 10 章 自定义数据类型	85	12.4 实验习题	112
10.1 实验目的	85	第 13 章 C++输入/输出流类库	118
10.2 知识要点	85	13.1 实验目的	118
10.3 实验范例	86	13.2 知识要点	118
10.3.1 结构体的基本应用	86	13.3 实验范例	118
10.3.2 结构体在函数调用中的应用	88	13.3.1 数据文件写入的应用	118
10.4 实验习题	89	13.3.2 数据文件读取的应用	120
第 11 章 类与对象	95	13.4 实验习题	122
11.1 实验目的	95	第 14 章 异常处理	126
11.2 知识要点	95	14.1 实验目的	126
11.3 实验范例	97	14.2 知识要点	126
11.3.1 构造函数和析构函数的应用	97	14.3 实验范例	127
11.3.2 函数模板的应用	99	14.4 实验习题	129
11.4 实验习题	100	第 15 章 图形界面编程	131
第 12 章 继承与多态	107	15.1 实验目的	131
12.1 实验目的	107	15.2 知识要点	131
12.2 知识要点	107	15.3 实验范例	132
12.3 实验范例	107	15.3.1 对话框应用程序	132
12.3.1 派生类的基本应用	107	15.3.2 单文档应用程序	135
12.3.2 运算符重载的基本应用	110	15.4 实验习题	137
第二部分 综合实践项目			
第 16 章 工程和数值计算类实践项目	141	第 18 章 游戏博弈类实践项目	161
16.1 解线性方程组	141	18.1 纸牌游戏	161
16.2 求函数积分	144	18.2 军棋博弈	177
16.3 扩展题目	147	18.3 扩展题目	185
第 17 章 信息处理类实践项目	149	第 19 章 其他综合类实践项目	187
17.1 单词复习	149	19.1 绘制函数图形	187
17.2 信息加密	155	19.2 鼠标操作六子棋	190
17.3 扩展题目	158	19.3 扩展题目	196
附录	197		
附录 A 常用流程图符号	197		
附录 B 常见错误信息和解决方法	198		
附录 C 实验报告格式	199		
附录 D 综合实践项目报告格式	201		
附录 E 主教材习题参考答案	203		
参考文献	234		

基本框架

第一部分 上机实验

第 1 章 集成开发环境与程序基本框架

1.1 实验目的

- (1) 熟悉 Visual C++ 6.0 的开发环境。
- (2) 掌握用 Visual C++ 6.0 编写标准控制台程序。
- (3) 熟悉 Code::Blocks 10.05 的开发环境。
- (4) 掌握用 Code::Blocks 10.05 编写标准控制台程序。
- (5) 了解使用断点或单步执行的方法进行程序调试。

1.2 知识要点

1. Visual C++ 6.0 集成开发环境

用 C 或 C++ 语言编写的源程序代码需要输入计算机，并进行编译、连接、调试，生成可执行的文件，在计算机中运行才能得到运行结果。Visual C++ 6.0（简称 VC）就是可以实现以上目的集成开发环境，它既支持 C++ 语言，也兼容 C 语言。

1) 创建 C/C++ 语言程序文件

进入 VC 集成开发环境，选择菜单“文件”|“新建”命令，在打开的“新建”对话框中，选择“文件”选项卡中的 C++ Source File 选项，在“文件”文本框中输入要建立的源程序文件名，并选择保存路径，如图 1-1 所示。

注意：系统默认的文件扩展名是 C++ 语言的“.cpp”，如果建立的是 C 语言源程序，一定要使用“.c”作为文件的扩展名。

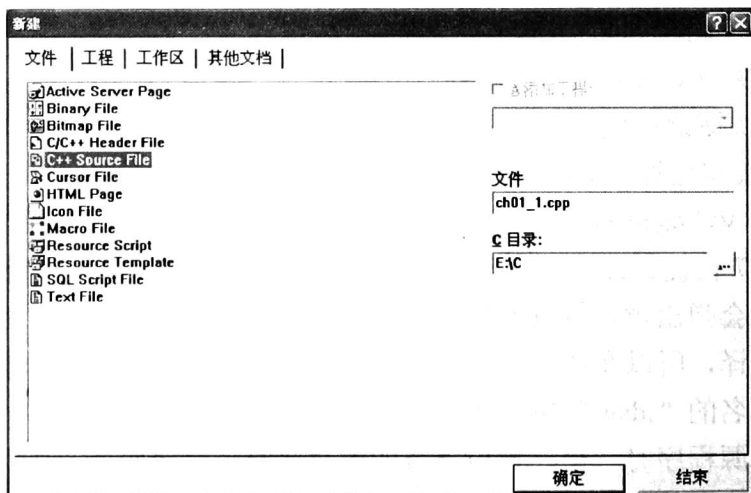


图 1-1 创建 C/C++ 语言程序文件

4 第 1 章 集成开发环境与程序基本框架

填好图 1-1 所示对话框的相关内容后，单击“确定”按钮，即可进入程序编辑页面，在右面的窗口中可以输入、编辑和修改程序代码，如图 1-2 所示。

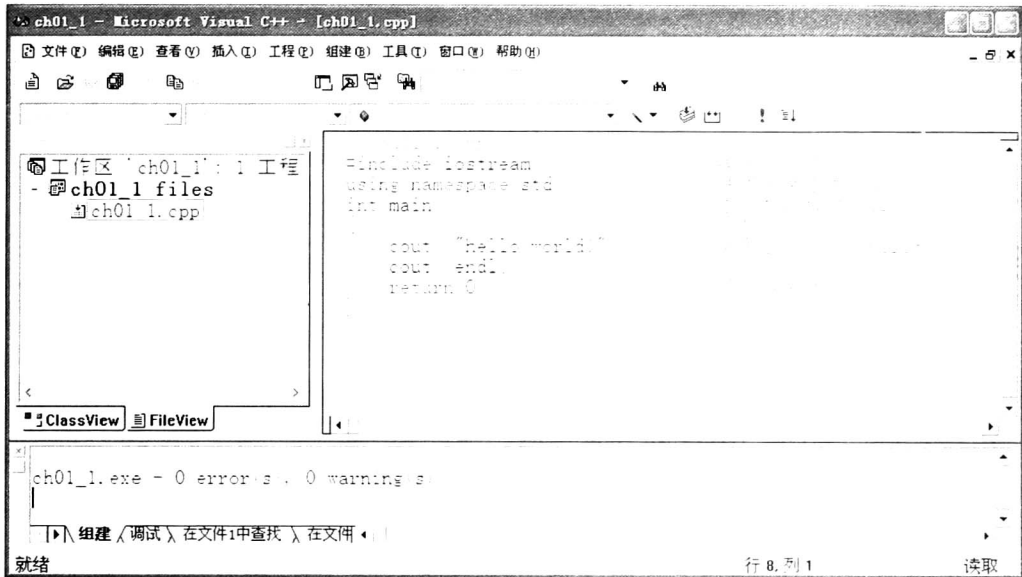


图 1-2 在 VC 中编辑 C++源程序

2) 编辑 VC 的程序代码

VC 编辑器是标准的 Windows 窗口程序，在这里可以输入和修改源程序代码，与 Windows 中的插入、删除、移动、复制和粘贴等功能的操作方法一致，而且还有许多专门为代码编写而开发的功能。例如：

- 关键字特殊颜色。
- 输入要调用的函数名称时，自动提示函数的参数。
- 输入对象名称及“.”时，自动提示对象的成员供选择。
- 按 Ctrl+] 快捷键自动寻找匹配的括号。
- 按 Ctrl+F2 快捷键可以定义书签，按 F2 键可以在选项卡快速移动光标。
- 按 Alt+F8 快捷键可以自动调整选定代码自动缩进等风格。
- 根据输入的代码，判断缩进或反缩进。
- 按 Ctrl+F7 快捷键可以编译源程序。
- 按 Ctrl+F5 快捷键可以运行程序。

3) 编译、连接和运行 C/C++源程序

将源程序输入 VC 编辑器后，需要将其编译成二进制代码。单击图 1-2 中的“编译”按钮或者按 Ctrl+F7 快捷键或者选择菜单“组建”|“编译”命令可以进行代码的编译。程序第一次编译时，会弹出对话框询问是否建立一个默认的项目工作区。在 VC 中必须先建立项目才能开始编译，所以在此单击 Yes 按钮。这样会在保存“.cpp”或者“.c”源程序文件的目录中生成同名的“.dsw”和“.dsp”等文件。

在 VC 中编译源程序时，会在窗口下方的消息区显示编译报告。如果编译出错，消息区内也会显示出现错误的代码位置和可能的出错原因，双击错误提示信息，光标会跳转到发生错误的代码行。

编译完成后，会在源程序文件保存的目录下生成同名的“.obj”目标文件。这时，还需要将目标文件连接生成可执行文件。单击图 1-2 中的“连接”按钮或者按 F7 键或者选择菜单“编译”|“构件”命令即可对目标文件进行连接，一般会生成同名的“.exe”文件。

如果编译和连接提示错误，VC 会提供出错位置和错误类型等出错信息，可以根据分析、修改出错代码，然后重新编译、连接。

如果编译、连接都没有错误，程序就可以运行了。单击图 1-2 中的“运行”按钮或者按 Ctrl+F5 快捷键或者选择菜单“编译”|“执行”命令，程序会在一个新打开的 DOS 窗口中运行并显示结果，如图 1-3 所示。

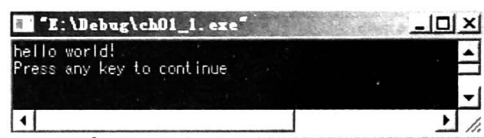


图 1-3 VC 中的程序运行结果

程序运行结果的后面会显示：Press any key to continue。这是 VC 自动加上的，并不是程序的实际输出。显示此提示信息时，说明程序已经正常运行完毕，可以按任意键关闭窗口。

4) 调试 C/C++ 源程序

程序中的错误可以通过调试的方式查找位置和原因，一般可以采取单步执行和断点两种方式。

(1) 单步执行。通过单步执行，可以逐条语句或逐个函数执行程序，每执行完一条语句或一个函数，程序就会暂停，用户可以检查每个语句或每个函数的执行结果。

按 F10 键，程序可以实现单步执行，这时可以在左下角的变量窗口观察变量的建立和值的情况，如图 1-4 所示。

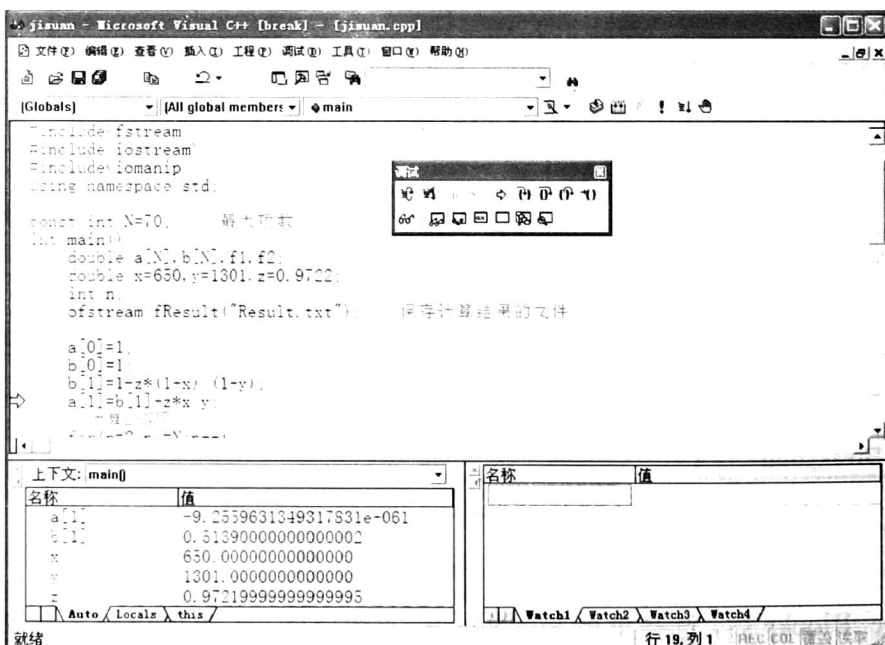


图 1-4 “单步执行”窗口

(2) 设置断点。可以让程序在任何地方设置断点，执行到断点时程序会暂停运行，方便检查和分析程序。一个程序可以设置多个断点。

把光标置于要设置断点的行，按 F9 键，这行代码的前面就会出现一个红色的大圆点，标志着断点已设置。在该行再次按 F9 键可以取消断点。按 F5 键可以开始调试程序，运行到断点处程序暂停，进入跟踪状态，并在断点处显示黄色的向右箭头，如图 1-5 所示。再次按 F5 键程序会继续运行。

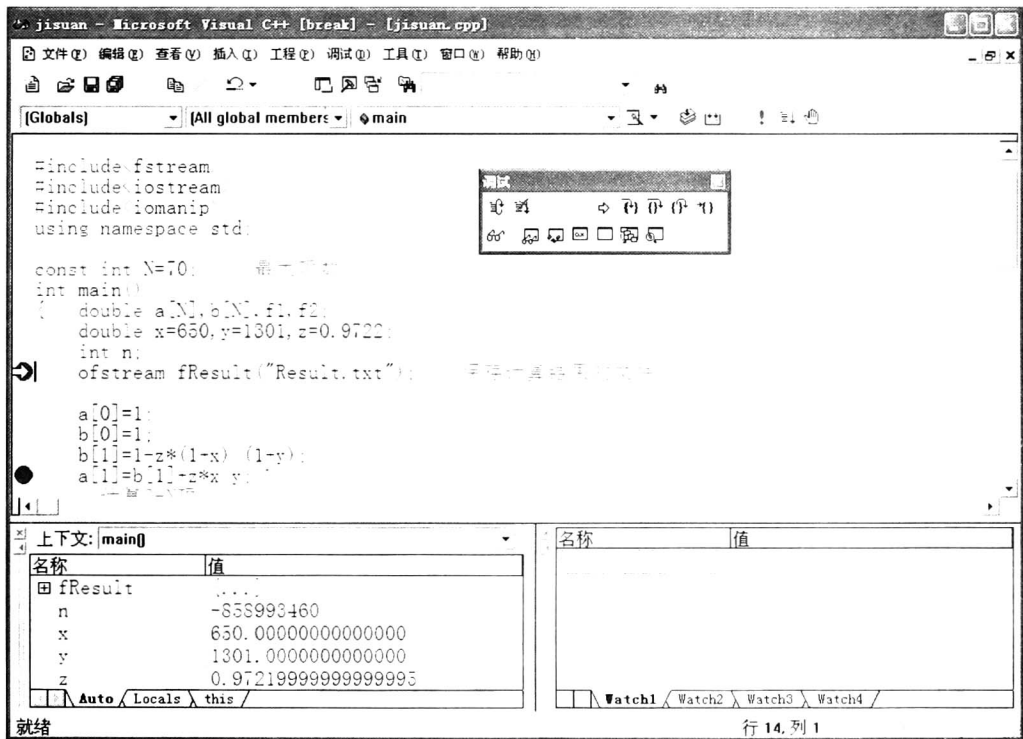


图 1-5 断点跟踪

VC 在调试菜单下常用的调试命令见表 1-1。

表 1-1 VC 调试命令

菜单命令	快捷键	说明
Go	F5	从当前语句开始或继续执行程序，直到程序结束或断点
Stop Debugging	Shift + F5	停止调试程序
Step Into	F11	单步进入，遇到函数则进入函数内部执行
Step Over	F10	单步执行，将所调用的函数作为整体来执行
Step Out	Shift + F11	跳出当前函数
Run to Cursor	Ctrl + F10	运行到光标所在行
Insert/Remove Breakpoint	F9	插入或删除断点

2. Code::Blocks 集成开发环境

1) 创建控制台应用程序

启动 Code::Blocks 后，选择菜单 File|New|Project 命令，或者在 Start here 页面中单击链接 Create a new project。系统弹出 New from template（新建项目）对话框，如图 1-6 所示。

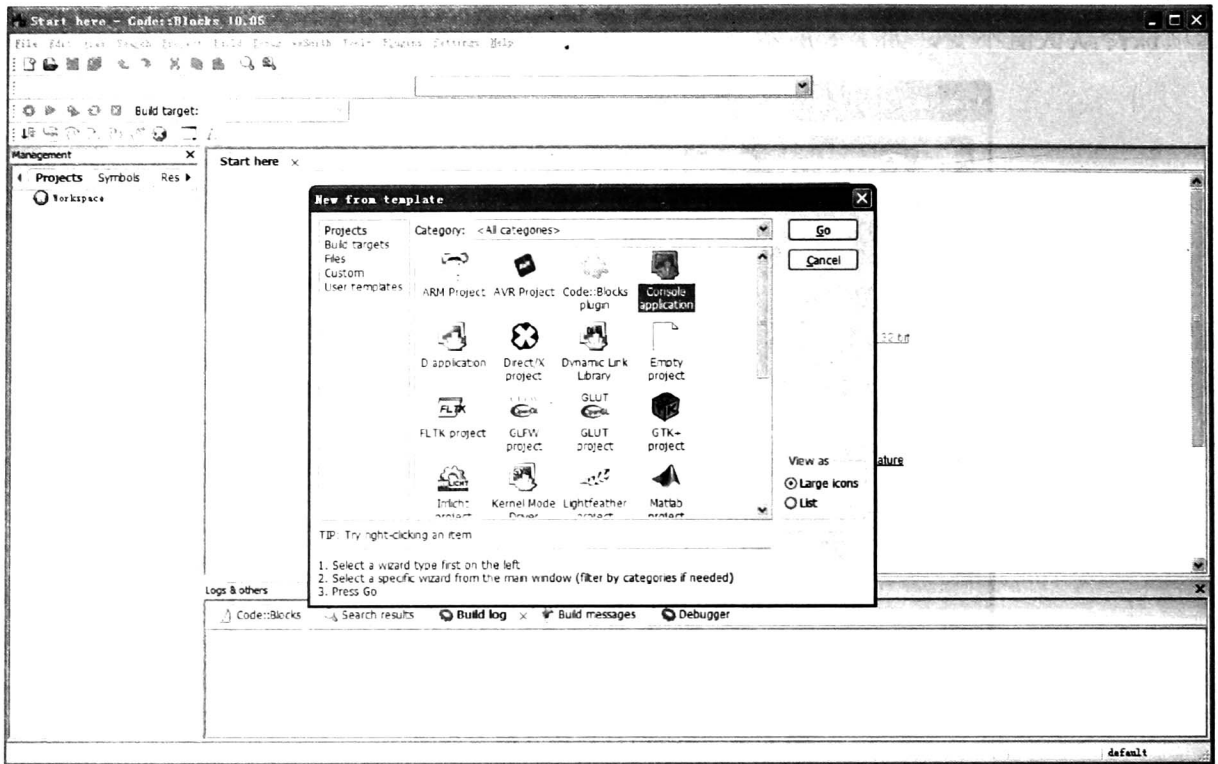


图 1-6 新建项目对话框

在如图 1-6 所示的对话框中，选择 Console application（控制台应用程序）选项，然后单击 Go 按钮，进入控制台应用程序向导。在 Please make a selection 列表框中选择要创建的是 C 语言还是 C++ 语言源程序，如图 1-7 所示。

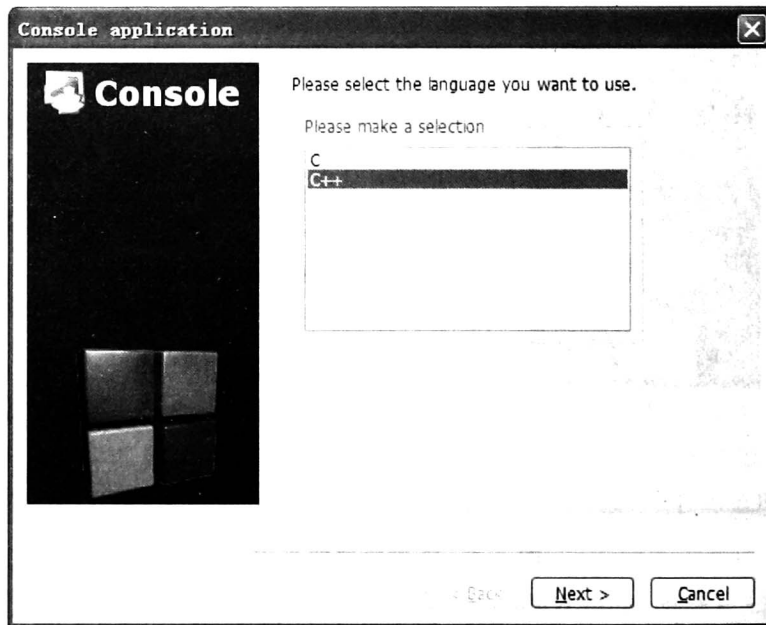


图 1-7 选择语言类型

C/C++ 语言选择完成后，单击 Next 按钮，在对话框中的 Project title（项目名称）文本框中输入“HelloWorld”，在 Folder to create project in 文本框中选择项目保存的位置，其他

选项保持默认值不变，如 1-8 所示。

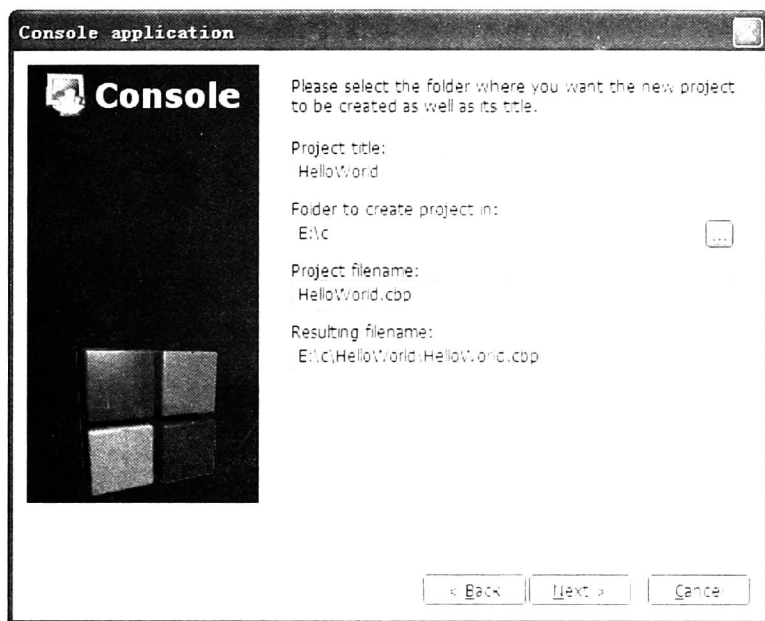


图 1-8 确定项目名称及保存位置

输入项目名称并确定项目保存的位置后，单击 Next 按钮，选择编译器类型，如图 1-9 所示，单击 Finish 按钮，结束向导。

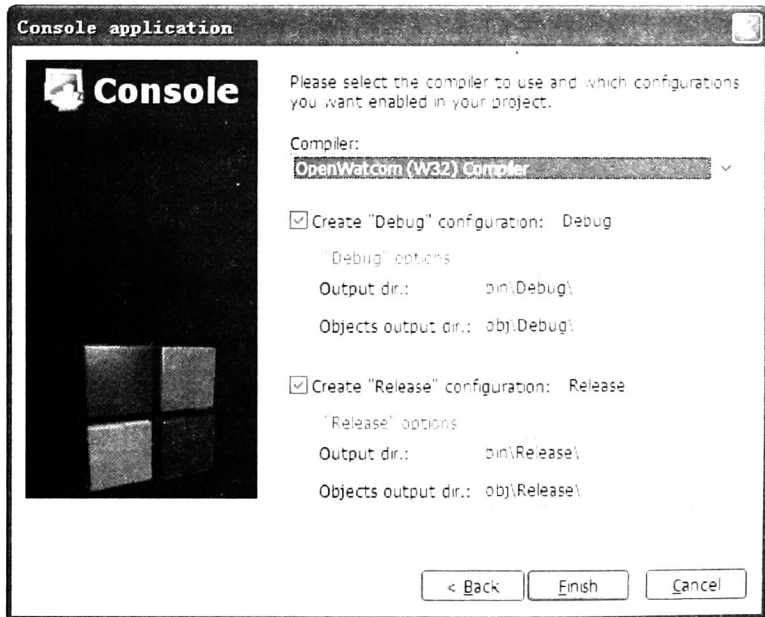


图 1-9 选择编译器类型

创建项目完成后，进入 Code::Blocks 程序代码编辑界面。左侧为项目浏览界面，双击 HelloWorld|Sources|main.cpp，系统会显示一个显示字符串“Hello world!”的程序，如图 1-10 所示。在 Code::Blocks 中编辑程序代码时，字体大小可以通过按 Ctrl+“鼠标中间键”（滚动）进行调整。

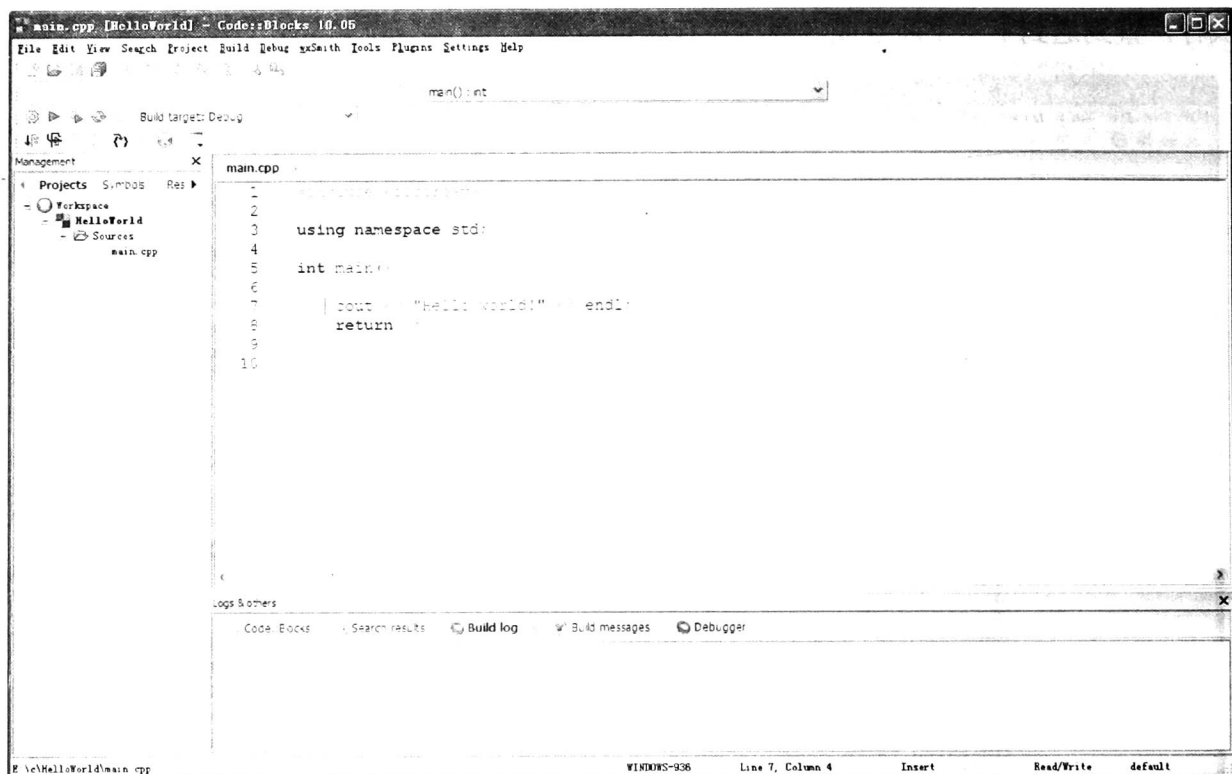


图 1-10 Code::Blocks 程序代码编辑界面

2) 编译、连接和运行程序

(1) 编译源程序。选择菜单 **Build|Compile current file** 命令或者按 **Ctrl+Shift+F9** 快捷键可以编译源程序，生成目标文件。

(2) 生成可执行文件。选择菜单 **Build|Build** 命令或者按 **Ctrl+F9** 快捷键可以编译源程序并连接目标文件，生成可执行文件。

(3) 重新生成可执行文件。生成可执行文件后，如果源程序没做任何修改，选择菜单 **Build|Build** 命令或者按 **Ctrl+F9** 快捷键，Code::Blocks 会忽略此操作。如果希望重新生成可执行文件，选择菜单 **Build|Rebuild** 命令或者按 **Ctrl+F11** 快捷键可以编译源程序并连接。

(4) 运行程序。选择菜单 **Build|Build Run** 命令或者按 **F9** 键。

3) 调试程序

(1) 设置断点。在 Code::Blocks 代码编辑窗口中，在代码行号右侧的空白处单击，或者在光标所在行按 **F5** 键，就会出现红色的圆点，表示在该行设置了断点，如图 1-11 所示。

(2) 开始调试。设置断点后，就可以开始进行断点调试了。选择菜单 **Debug|Start** 命令，或者按 **F8** 键开始调试。程序会在断点处暂停运行，这时在断点处红色圆点内会有一个小黄色的小三角。

(3) 观察变量。选择菜单 **Debug|Debugging Windows|Watches** 命令，启动变量观察窗口，可以观察各个变量的当前值。

(4) 调试工具。Code::Blocks 调试工具栏如图 1-12 所示。

该工具栏从左至右各按钮的功能如下。

Debug/Continue: 开始和继续调试。如果程序暂停在断点处，单击此按钮程序会继续

执行，直到遇到下一个断点或者程序执行结束。

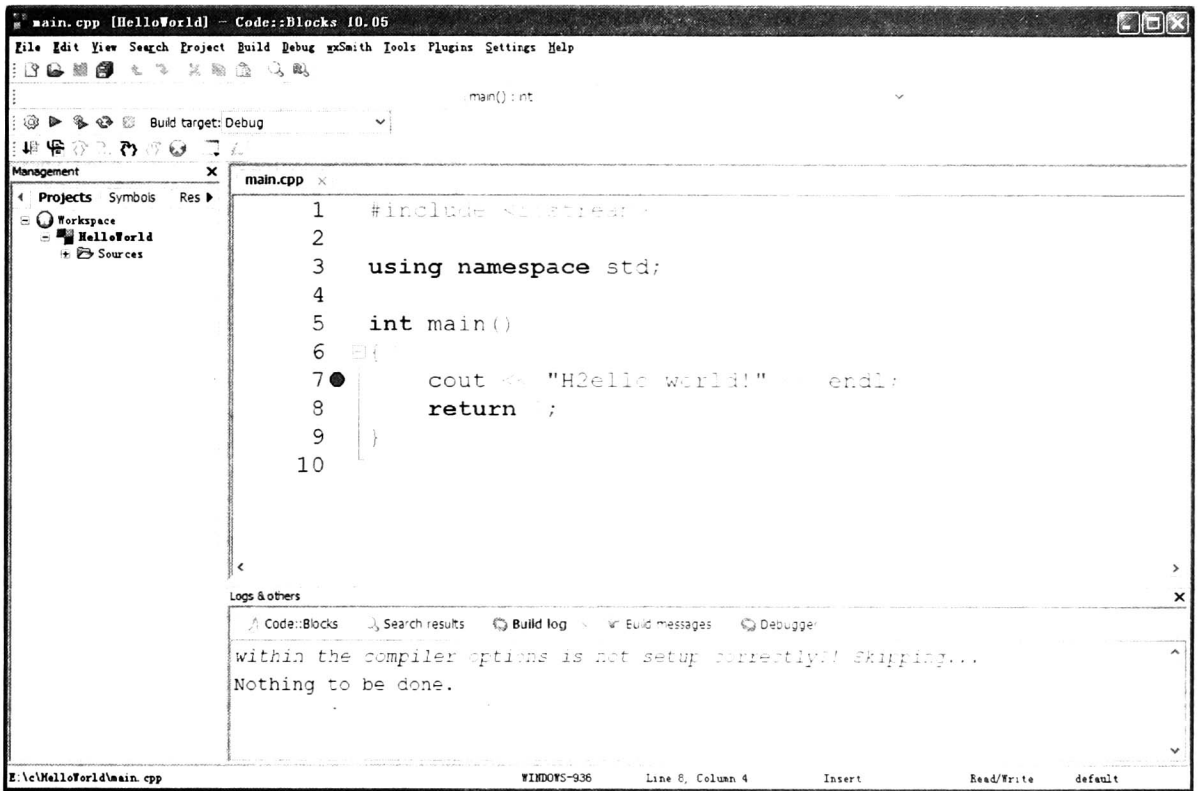


图 1-11 在 Code::Blocks 中设置程序断点



图 1-12 Code::Blocks 的调试工具栏

Run to cursor: 执行程序到光标所在处中断。

Next line: 执行下一行代码，然后在下一行中断。即使本行含有函数调用，也不会进入函数执行，而是直接跳过该函数。

Next instruction: 执行下一条指令，即下一条汇编语言。该功能不常用。

Step into: 与 Next line 功能相近，不同的是此功能会进入函数内部执行。

Step out: 跳出。跳出正在执行的函数，返回主调函数。

Stop debugging: 中止调试。

Debugging windows: 观察窗口，可以查看变量、CPU 的寄存器状态等。

Various into: 程序执行时的相关信息窗口。

1.3 实验范例

1. 问题的提出

一个学生参加了 3 门课程的考试，成绩分别为语文 90 分、数学 85 分、物理 90 分。计算该学生的平均成绩。

2. 算法及程序设计分析

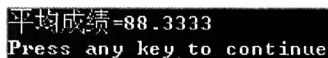
- (1) 定义保存各科成绩和平均成绩的变量: chinese、math、physics、average。
- (2) 令 90→chinese、85→math、90→physics, 即 chinese=90、math=85、physics=90。
- (3) 计算 (chinese+math+physics) /3→average。
- (4) 显示平均成绩 average。

3. 源程序及说明

```
//ex01_1.cpp
#include<iostream>
using namespace std;
int main( )
{   int chinese,math,physics;    //定义整型变量, 保存各科成绩
    float average;              //定义单精度变量(实数), 保存平均成绩
    chinese=90;                  //语文成绩
    math=85;                     //数学成绩
    physics=90;                  //物理成绩
    average =(chinese+math+physics)/3.0;    //计算平均成绩
    cout<< "平均成绩="<< average<<endl;    //显示平均成绩
    return 0;
}
```

4. 程序运行结果

将以上代码输入 C++开发环境并运行, 结果如图 1-13 所示。



```
平均成绩=88.3333
Press any key to continue
```

图 1-13 程序运行结果

思考: 如果将语句 `average =(chinese+math+physics)/3.0` 中的除数 3.0 换成 3, 结果会如何?

1.4 实验习题

1. 阅读程序, 写出下面程序的执行结果, 再上机运行, 观察基本程序结构执行效果。

```
//ex01_2.cpp
//This is a C++ program.
#include<iostream>
using namespace std;
int main( )
{
    cout<<34567<<endl;
```