



育本科国家级规划教材

课程配套教材

辽宁省精品课程配套教材

高等学校计算机基础教育规划教材

程序设计基础（C语言） 实验指导与测试（第3版）

高克宁 等 编著



清华大学出版社



普通高等教育本科国家级规划教材

教育部—英特尔精品课程配套教材

辽宁省精品课程配套教材

高等学校计算机基础教育规划教材

程序设计基础（C语言） 实验指导与测试（第3版）

高克宁 李金双 焦明海 张昱 编著

清华大学出版社
北京

内 容 简 介

本书是《程序设计基础(C语言)(第3版)》的配套实验与测试教材,全书分为3个部分,分别是实验指导、基本概念测试和工程案例。其中实验部分和基本概念测试部分是配合《程序设计基础(C语言)(第3版)》中各章节教学内容所安排的。实验部分具有覆盖相应章节教学内容、突出各知识点、实验指导细致的特点。基本概念测试中提供了与教材中各章节相对应的测试题,以利于学习者加深理解,拓宽知识,提高能力。

本书适合作为高等院校理工类各专业本科生教材,也可作为计算机培训教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

程序设计基础(C语言)实验指导与测试 / 高克宁等编著. —3版. —北京:清华大学出版社, 2018
(高等学校计算机基础教育规划教材)

ISBN 978-7-302-49853-7

I. ①程… II. ①高… III. ①C语言—程序设计—高等学校—教学参考资料 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 052435 号

责任编辑:袁勤勇 战晓雷

封面设计:常雪影

责任校对:白蕾

责任印制:杨艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:19.5 字 数:444千字

版 次:2013年8月第1版 2018年5月第3版 印 次:2018年5月第1次印刷

印 数:1~1500

定 价:49.00元

产品编号:076097-01

前言

程序设计是一门实践性很强的课程,仅仅通过理论学习不足以完全掌握程序设计的精髓,必须通过大量的程序设计实践来提高对程序设计的认知。本书作为《程序设计基础(C语言)(第3版)》的配套教材,旨在帮助学生掌握程序设计的基本技能。

本书共分3个部分。第1部分以跨平台开发环境 Code::Blocks 为实验环境,针对教材精心设计了14个实验,每个实验包括实验目的、实验指导和实验内容。其中实验指导给出了详细的实验设计思路和实验步骤,对自行开展实验活动具有重要的指导意义。第2部分针对教材各章的内容相应地给出了大量的基本概念测试题,对加深和巩固对基本概念的理解有很大帮助。第3部分给出了两个综合案例,每个案例都按照程序设计的完整过程给予详尽的指导,以进一步培养学生的综合实践能力。附录主要包括与 Code::Blocks 实验环境对应的 Visual C++ 6.0 集成开发环境介绍、实验内容奇数题参考答案、基本概念测试参考答案及解析和常用 C 语言函数库。

参与本书编写的主要人员有高克宁、李金双、焦明海、张昱、李凤云、李婕、赵长宽等。

本书是“十二五”普通高等教育本科国家级规划教材、教育部-英特尔精品课程配套教材、辽宁省精品资源共享课程配套教材。

由于作者水平有限,书中难免有不足之处,真诚地欢迎各位专家和读者批评指正,以帮助我们进一步完善本书。作者的联系方式如下:

电子邮件: gkn@cc.neu.edu.cn

通信地址: (110819)辽宁 沈阳 东北大学计算中心 高克宁

作 者

2018年1月于东北大学

目录

第1部分 实验指导

实验 1	熟悉实验环境	3
实验 2	简单程序设计	11
实验 3	分支控制结构	15
实验 4	循环控制结构	19
实验 5	数组与字符串	29
实验 6	函数	33
实验 7	指针	38
实验 8	结构体	49
实验 9	预编译和宏定义	55
实验 10	文件	59
实验 11	程序设计思想及范例	64
实验 12	面向对象程序设计	72
实验 13	并行程序设计	77
实验 14	个体软件开发	91

第2部分 基本概念测试

测试 1	计算机及程序设计概述	97
测试 2	信息编码与数据类型	99
测试 3	基本运算与顺序结构	101
测试 4	逻辑判断与选择结构	105
测试 5	迭代计算与循环结构	113
测试 6	集合数据与数组	120
测试 7	模块化与函数	125
测试 8	地址操作与指针	133
测试 9	复杂数据类型与结构体	141
测试 10	泛化编程与预编译	150
测试 11	数据存储与文件	157

测试 12	程序设计思想及范例	167
测试 13	面向对象及 C++ 基础	173
测试 14	并行程序设计	175
测试 15	个体软件开发	179

第 3 部分 工程案例

案例 1	万年历	185
案例 2	通讯录	203
附录 A	Microsoft Visual C++ 6.0 实验环境	214
附录 B	实验指导奇数题参考答案	233
附录 C	基本概念测试参考答案及解析	263
附录 D	常用 C 语言库函数	297

第1部分

实验指导

熟悉实验环境

Code::Blocks 是一个开放源码的全功能跨平台 C/C++ 集成开发环境,可稳定运行在 32 位和 64 位操作系统中。附录 A 提供了另一个开发软件——Microsoft Visual C++ 6.0 的介绍,请读者根据自己的实际情况选择合适的软件进行编程实践。在各实验的“实验指导”部分给出了必要的提示信息。

1.1 实验目的

- (1) 熟悉 C 语言运行环境,了解和使用 Code::Blocks 集成开发环境。
- (2) 熟悉 Code::Blocks 环境的功能键和常用的功能菜单命令。
- (3) 掌握 C 语言程序的书写格式和 C 语言程序的结构。
- (4) 掌握 C 语言上机步骤,以及编辑、编译和运行一个 C 程序的方法。
- (5) 掌握 Code::Blocks 环境下的简单编译错误的修改方法。

1.2 实验指导

在本实验中,使用 Code::Blocks 编制两个简单的程序。程序的要求和目标如下:

(1) 在屏幕上输出一个字符串“Hello World!”,掌握编辑、编译、运行一个 C 程序的方法。

(2) 用键盘输入两个数,计算并输出这两个数的和。通过对该程序的修改运行,初步掌握 Code::Blocks 中的简单编译错误的修改方法。

如果读者使用 Microsoft Visual C++ 6.0 软件进行实验操作,下面的实验指导部分请参考附录 A 的“A.1 熟悉 Microsoft Visual C++ 6.0 实验环境”。

1. 程序一

编辑、编译和运行 C 语言程序的过程主要包括以下 4 个步骤:

- 编辑:将程序代码输入 C 程序源文件(.c 文件)中。
- 编译:将源文件编译成目标程序文件(.obj 文件)。

- 链接：将目标程序文件和其他相关文件链接成可执行文件(.exe 文件)。
- 运行：运行可执行文件。

上述 4 个步骤中,第一步的编辑工作是最繁杂的,必须由编程人员在编辑器中逐步编写完成,其余几个步骤则相对简单,基本上由集成开发环境自动完成。

启动 Code::Blocks 应用软件,在如图 1-1 所示的编译器自动检测对话框中选择 GNU GCC Compiler。GCC 是一种能应用到许多操作系统上的编译器,我们用它编译 C 语言。单击 OK 按钮确定所做的选择。

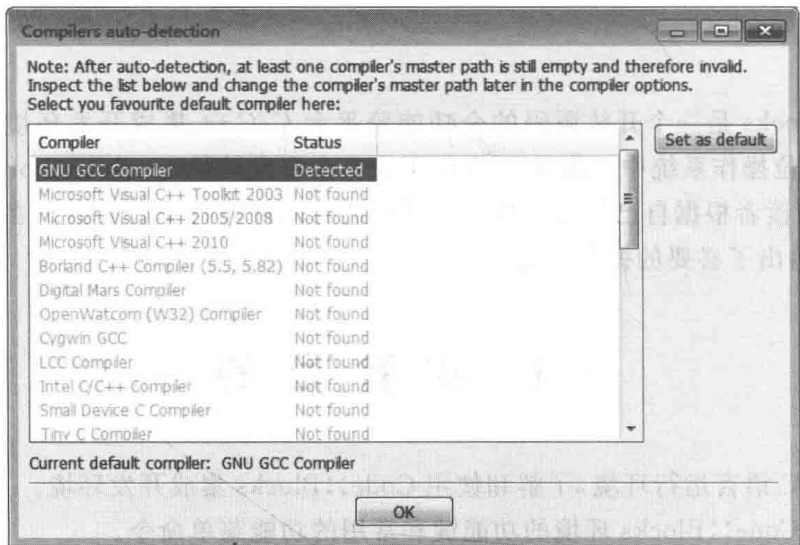


图 1-1 编译器自动检测对话框

在如图 1-2 所示的 Code::Blocks 应用程序界面中,选择 File 菜单中的 New→Project 命令。

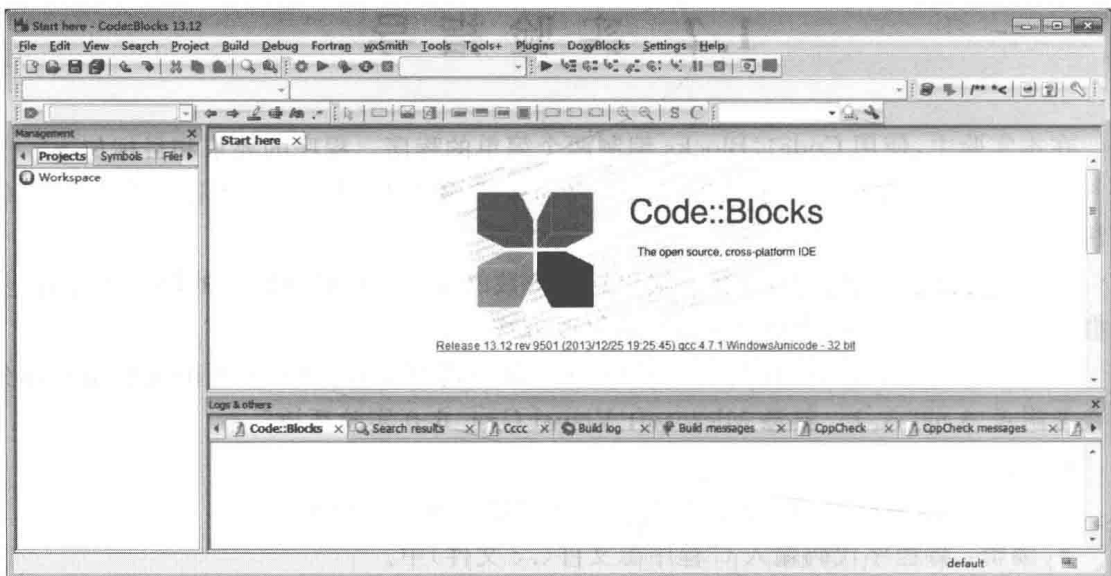


图 1-2 Code::Blocks 应用程序界面

弹出图 1-3 所示的 New from template 模板选择对话框,选择 Console application,单击 Go 按钮继续。

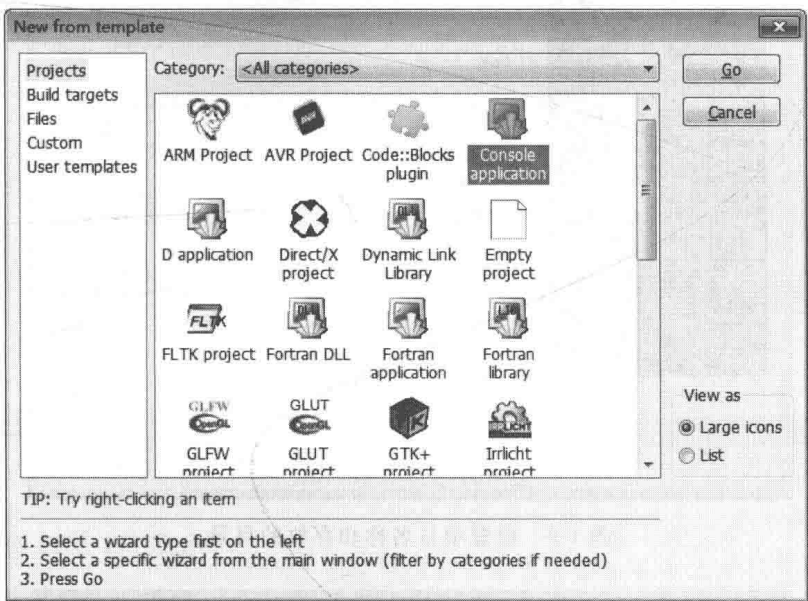


图 1-3 模板选择对话框

在弹出的如图 1-4 所示的 Console application 对话框中选择 C 语言,单击 Next 按钮继续。

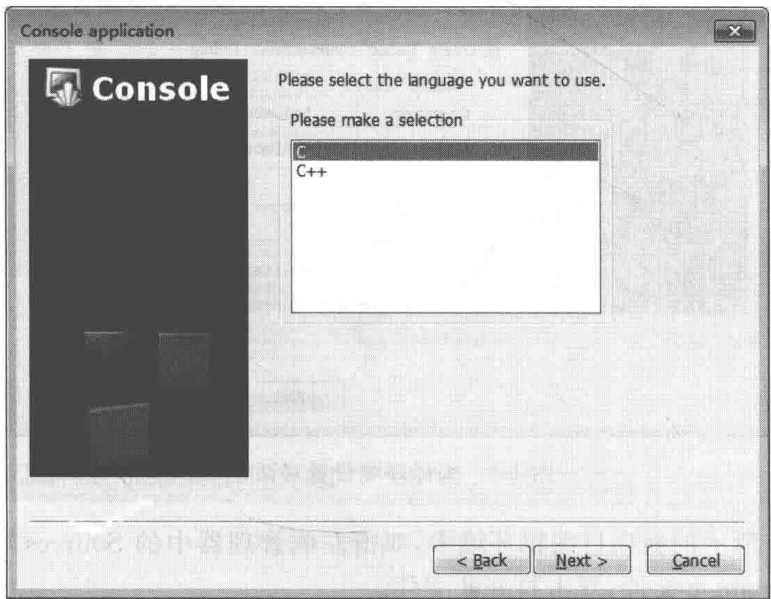


图 1-4 选择语言

在弹出的如图 1-5 所示的 Console application 对话框的第二个页面中设定项目名称为 hello,项目存放的目录为“d:”,单击 Next 按钮继续。

在弹出的如图 1-6 所示的 Console application 对话框的第三个页面中设置编译环境,并单击 Finish 按钮完成新项目的创建工作。

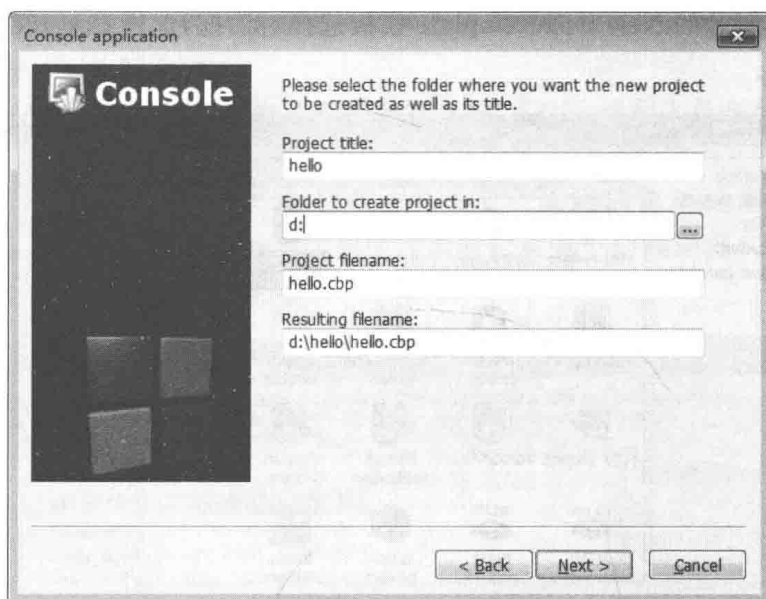


图 1-5 设置项目名称和存放的目录



图 1-6 编译环境设置对话框

在如图 1-7 所示的新项目编辑环境中,双击左侧管理器中的 Sources 项,显示系统已经生成了一个 main.c 文件,双击打开此文件。

在右侧显示出系统已生成的简单程序代码。删除这些代码,然后输入下面的代码:

```
#include <stdio.h>
void main()
{
    printf("Hello World!\n");
}
```

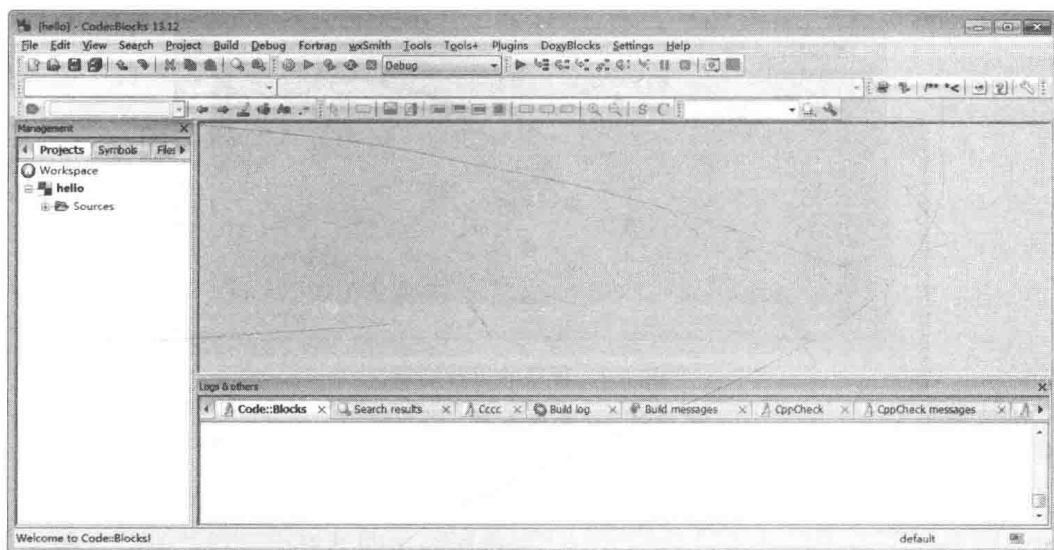


图 1-7 新项目编辑环境

完成后界面如图 1-8 所示。

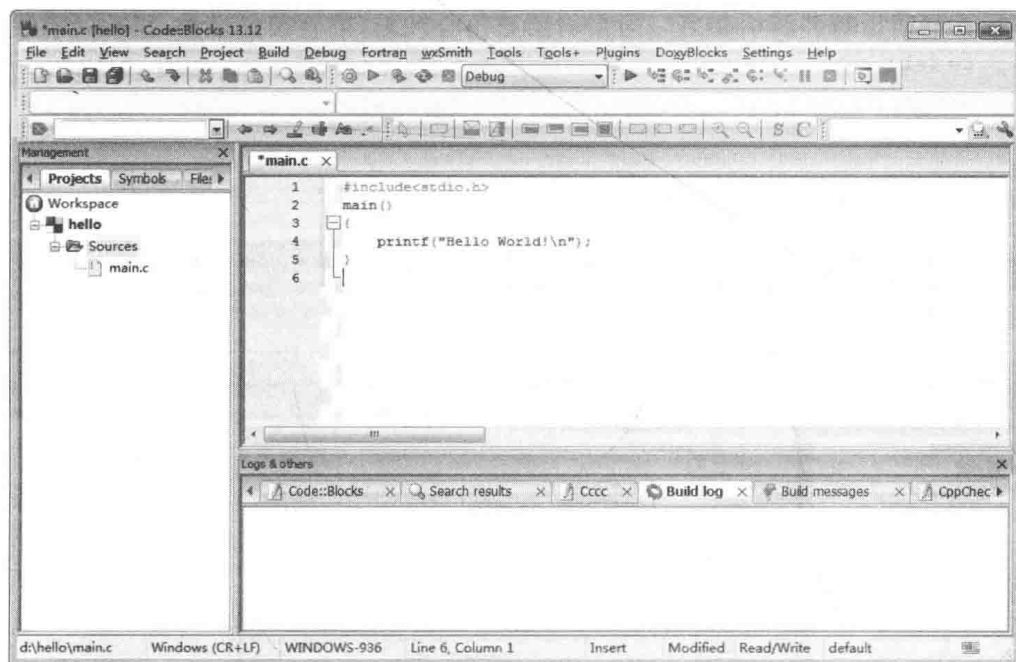


图 1-8 编辑源文件

选择 Build 菜单下的 Build and run 命令创建并运行程序,显示如图 1-9 所示的程序运行结果,即在控制台窗口中显示字符串“Hello World!”。

按下键盘任意一个按键后,此控制台界面关闭。

2. 程序二

编写加法程序,输入两个整数,输出它们的和。

在开始编写第二个程序时,应建立新的空工程,创建新的 C 语言源文件,工程的名称

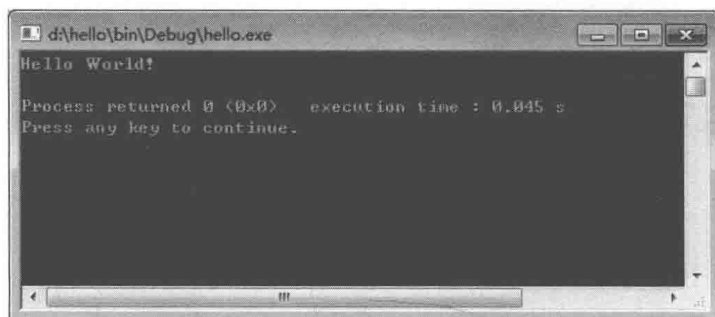


图 1-9 程序的运行结果

为 add。

加法程序代码如下：

```
#include <stdio.h>
main()
{
    int a,b,c;
    scanf("%d%d",&a,&b);
    c=a+b;
    printf("%d\n",c);
}
```

源程序编辑界面如图 1-10 所示。

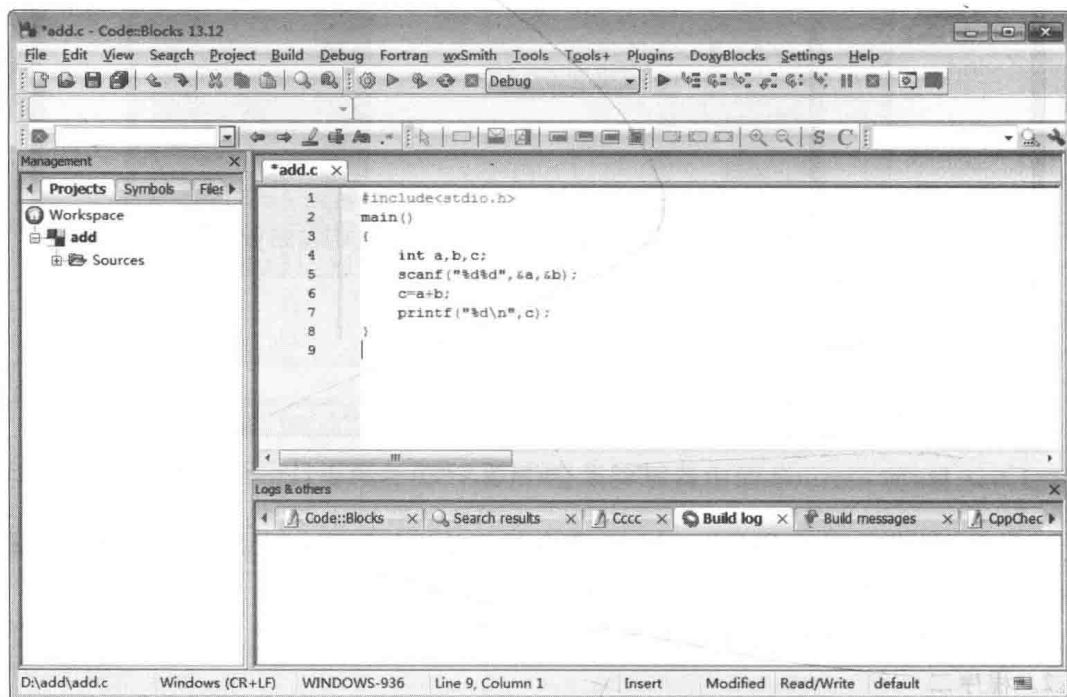


图 1-10 加法程序的源代码

创建并运行程序,在程序运行窗口中输入 3 5(表示空格,下同),然后回车,运行结果如图 1-11 所示。

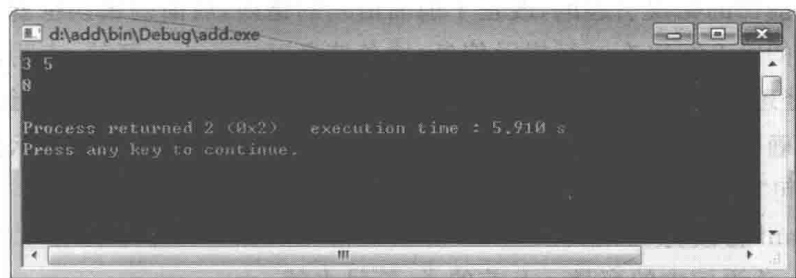


图 1-11 加法程序的运行结果

另一种输入方式是输入 3 后回车,再输入 5 后回车,也能得到正确的结果。可输入“3,5”,看一下程序的运行结果。

提示:如果在编写第二个程序时,没有关闭原来的工程而直接创建新的 C 语言源文件,程序编译运行的就可能是原来的程序,而不是新创建的源程序。

在程序运行正确的基础上,删除“scanf(“%d%d”,&a,&b);”语句以及“c=a+b;”语句末尾的分号,重新编译程序,这时会出现编译错误。在编译器的输出窗口显示有编译错误,如图 1-12 所示。

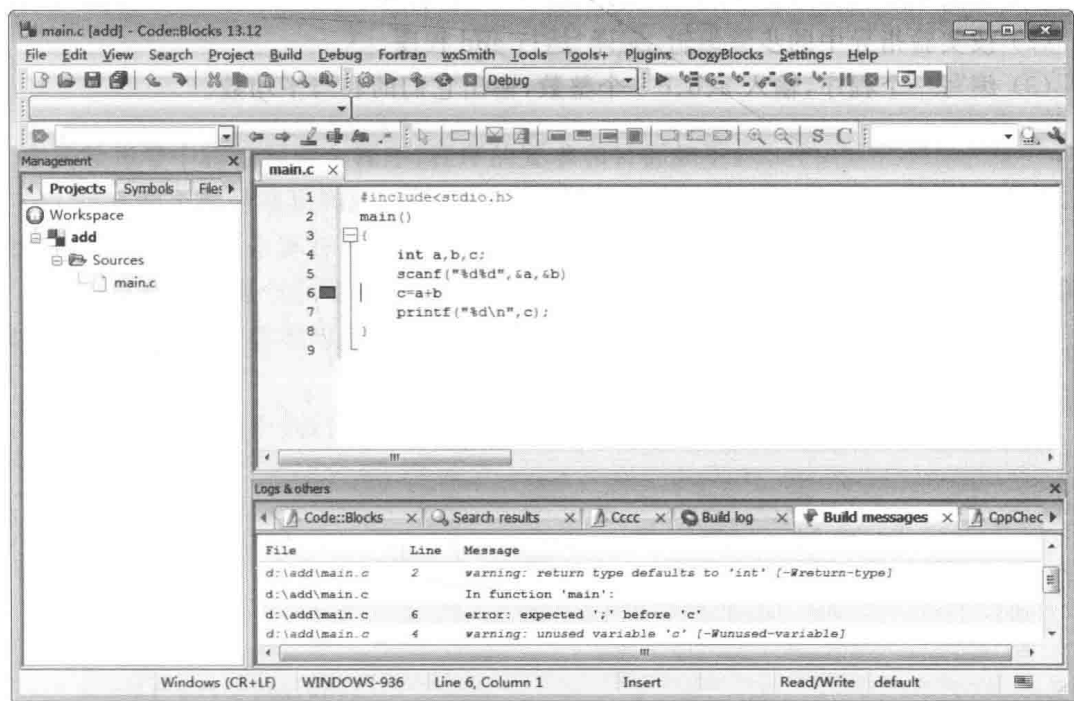


图 1-12 编译错误提示界面

在源程序的第 6 行出现红色方块标记,在下面的 Build messages 中提示信息为“error: expected ';' before 'c'”。因为在源程序中可任意输入空格和回车,因此在标识符 c 前面添加分号实际上就是在 scanf 语句后添加分号。

需要注意的是,只有在本例这样明确可知下一条语句后也应添加分号的情况下,才可一次修改若干个错误。通常情况下,改正第一条错误后就应该重新编译程序。有时后面的错误是由于前面的错误引起的,改正了前面语句的错误,后面的错误也就自动消失了。

在源程序的第 2 行也有一行提示信息,这条信息以 warning 开始,表明这是一条警示信息,不进行处理也不会影响程序的运行结果,在此不予处理,后面会有详细讲解。

在程序正确的情况下,分别修改下面的语句,熟悉简单的编译错误提示信息:

- (1) 将语句“int a,b,c;”中的 c 去掉。
- (2) 将 scanf 修改为 scamf。
- (3) 将 scanf 语句中的变量 a、b 前的 & 符号去掉。
- (4) 将 scanf 语句中的右引号去掉。
- (5) 在 printf 语句中的 c 前加上 & 符号。

上面有一些修改会产生编译错误,有一些修改会导致运行时得不到正确的结果(逻辑错误)。

1.3 实验内容

- (1) 按实验指导中的步骤编辑、编译、运行 hello 程序。
- (2) 按实验指导中的步骤编辑、编译、运行 add 程序。
- (3) 编写一个程序,输入 a、b、c 3 个整数,输出它们的和与平均数。

简单程序设计

2.1 实验目的

- (1) 掌握 C 语言数据类型,熟悉如何定义一个整型、字符型和实数类型的变量,以及不同的数据类型的常用输入输出方法。
- (2) 学会使用 C 语言的算术运算符,以及用这些运算符编写简单的算术表达式。
- (3) 能够编写简单的顺序结构程序。

2.2 实验指导

本实验指导中将编写两个程序,程序的要求和目标如下:

- (1) 计算两个数的乘积和商,熟悉单精度数、双精度数的输入输出,理解整数除法、实数除法的不同,掌握 % 运算符的用法。
- (2) 用键盘输入两个小写字母,输出其 ASCII 码值和对应的大写字母,熟悉字符变量的输入输出,理解字符类型与整数类型之间的关系。

1. 程序一

编写程序,输入两个数,计算它们的乘积和商。

代码一:

```
#include <stdio.h>
main()
{
    int a,b,c,d;
    scanf("%d%d",&a,&b);
    c=a*b;
    d=a/b;
    printf("%d,%d\n",c,d);
}
```

编译运行,按下面的要求输入数值,观察程序运行结果。