

高等学校计算机基础教育教材精选

C语言程序设计案例教程

(第3版)

刘兆宏 温荷 王会 编著

清华大学出版社



高等学校计算机基础教育教材精选

C语言程序设计案例教程

(第3版)

刘兆宏 温荷 王会 编著

清华大学出版社
北京

内 容 简 介

这是一本面向广大初学者的 C 语言案例教材,全书共 10 章:第 1 章~第 3 章介绍程序设计与 C 语言的基础知识;第 4 章~第 8 章介绍数组、函数、指针、结构体与共用体、文件等重要内容;第 9 章~第 10 章分别采用指针、数组、单链表来开发“学生成绩管理系统”,通过案例的分析实现培养初学者运用 C 语言开发中小型项目的能力。针对初学者和自学读者的特点,本书力求做到深入浅出,将复杂的概念用简洁的语言娓娓道来。全书以项目为主线,基础性和实用性并重。项目贯穿全书,通过对项目的实现和讲解,使读者逐步具备利用 C 语言来开发应用程序的能力。本书可作为高等院校学习 C 语言课程的教材或培训学校的教材,也可作为自学者的参考书。

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

C 语言程序设计案例教程/刘兆宏,温荷,王会编著. —3 版. —北京:清华大学出版社,2017

(高等学校计算机基础教育教材精选)

ISBN 978-7-302-47313-8

I. ①C… II. ①刘… ②温… ③王… 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(2017)第 101904 号

责任编辑:谢 琛

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:刘海龙

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 装 者:保定市中华美凯印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.75

字 数:432 千字

版 次:2008 年 9 月第 1 版 2017 年 6 月第 3 版

印 次:2017 年 6 月第 1 次印刷

印 数:1~2000

定 价:39.50 元

产品编号:073431-01

前言

C 语言程序设计案例教程(第 3 版)

一、本书特色

这是一本面向广大初学者的 C 语言程序设计案例教材。本书的特色是深入浅出、案例丰富、项目导学、立体配套。

针对初学者和自学读者的特点,本书力求做到深入浅出,将复杂的概念用简洁的语言娓娓道来。

全书以项目为主线,基础性和实用性并重。本书不仅详细介绍 C 语言本身,而且介绍编程思想、编程规范、编程方法等实用开发技术。

项目贯穿全书,通过对项目的分析、实现和讲解,使读者逐步具备利用 C 语言来开发应用程序的能力。

二、内容摘要

第 1 章 C 语言程序设计基础。作为全书的开篇,通过几个非常简单的例子介绍 C 语言的结构特点、书写格式、输入输出函数以及如何用 Code::Blocks 实现 C 语言程序的运行等内容。

第 2 章 数据类型、运算符与表达式。主要介绍 C 语言的基本数据类型、常量和变量、运算符及由它们组成的表达式、运算符的优先级与结合性等。

第 3 章 控制结构。通过一系列典型的实例,逐步介绍了算法的基础知识、流程图的绘制及各种控制结构语句的使用。

第 4 章 数组。介绍数值数组和字符数组,并对简单的学生成绩程序进行分析和实现。

第 5 章 函数。介绍函数的概念、定义及函数的调用方式。重点通过完成学生成绩管理程序来运用函数知识。

第 6 章 指针。本章主要围绕指针是什么、指针有何用、如何应用、具体应用来展开。

第 7 章 结构体与共用体。介绍结构体和共用体的概念、结构体数组的使用、结构体指针的应用等内容,并通过利用结构体知识完成学生成绩管理程序的分析及实现。

第 8 章 文件。介绍基本的文件知识,主要介绍 C 语言读写文件的方法。

第9章 综合实训 1。采用指针数组的方式来开发“学生成绩管理系统”，主要通过案例的分析实现来培养读者运用 C 语言开发中小型项目的能力。

第10章 综合实训 2。采用单链表来开发“学生成绩管理系统”，让读者体会不同实现算法效率上的异同。

三、使用指南及相关说明

为方便教师备课，本书配有视频、电子教案(PPT 文件)、教学要点、考试样题等教学资料，可从清华大学出版社网站下载。

本书第 1、3、6、9 章由温荷编写，第 2、4、8、10 章由王会编写，第 5、7 章由刘兆宏编写。参加本书策划、组织和编写的还有郑莉教授、张应辉博士和张辉教授。全书由刘兆宏统稿，张辉教授负责审阅。

由于作者水平有限，书中难免有不妥之处，欢迎读者对本书内容提出意见和建议，我们将不胜感激。

作 者

2016 年 12 月

目录

C 语言程序设计案例教程(第 3 版)

第 1 章 C 语言程序设计基础	1
1.1 简单的 C 程序	1
1.1.1 一个简单的 C 程序	1
1.1.2 C 程序的结构特点	2
1.1.3 C 程序的书写格式	3
1.2 C 语言概述	4
1.2.1 C 语言的产生及发展	4
1.2.2 C 语言的特点	4
1.3 C 语言程序的实现	5
1.3.1 运行 C 程序的步骤和方法	5
1.3.2 Code::Blocks 集成开发环境的使用	6
1.4 输入与输出函数	9
1.4.1 标准格式输出函数 printf()	9
1.4.2 标准格式输入函数 scanf()	16
1.4.3 字符输出函数 putchar()	20
1.4.4 字符输入函数 getchar()	21
1.5 本章小结	22
习题	22
第 2 章 数据类型、运算符与表达式	24
2.1 C 语言的数据类型	24
2.2 常量与变量	25
2.2.1 常量	25
2.2.2 变量	27
2.3 C 语言的基本数据类型	28
2.3.1 整型数据	28
2.3.2 实型数据	30
2.3.3 字符型数据	31
2.3.4 数据类型转换	32

2.4	运算符与表达式	33
2.4.1	算术运算符与算术表达式	33
2.4.2	赋值运算符和赋值表达式	35
2.4.3	逗号运算符与逗号表达式	36
2.4.4	sizeof 运算符	37
2.4.5	运算符的优先级和结合性	37
2.4.6	案例分析: 学生的总分及平均分计算	38
2.5	本章小结	39
	习题	39
第3章	控制结构	42
3.1	算法	42
3.1.1	算法的概念	42
3.1.2	算法的特性	42
3.1.3	算法的描述	43
3.1.4	三种基本结构和改进的流程图	43
3.2	选择结构	44
3.2.1	if 语句	44
3.2.2	案例分析: 成绩等级判定 1	48
3.2.3	switch 语句	49
3.2.4	案例分析: 成绩等级判定 2	51
3.3	循环结构	52
3.3.1	for 循环	52
3.3.2	案例分析: 计算平均成绩 1	56
3.3.3	while 循环	57
3.3.4	案例分析: 计算平均成绩 2	61
3.3.5	do-while 循环	63
3.3.6	循环的嵌套	63
3.4	跳转语句	65
3.4.1	break 语句	66
3.4.2	continue 语句	67
3.4.3	goto 语句	68
3.4.4	exit 语句	68
3.5	案例分析: 学生成绩管理程序	68
3.6	本章小结	71
	习题	71

第 4 章 数组	74
4.1 一维数组	74
4.1.1 一维数组定义	75
4.1.2 一维数组元素的引用	76
4.1.3 一维数组的初始化	77
4.1.4 案例分析：冒泡排序	80
4.2 二维数组	82
4.2.1 二维数组的定义	82
4.2.2 二维数组元素的引用	82
4.2.3 二维数组的初始化	85
4.2.4 案例分析：简单学生成绩程序	87
4.3 字符数组	89
4.3.1 字符数组的定义	89
4.3.2 字符数组的初始化	89
4.3.3 字符数组的引用	90
4.3.4 字符串和字符串结束标志	90
4.3.5 字符数组的输入输出	91
4.3.6 字符串处理函数	93
4.3.7 案例分析 1：输入五个国家的名称按字母顺序排列输出	96
4.3.8 案例分析 2：将无符号整数 n 翻译成 $d(2 \leq d \leq 16)$ 进制表示的字符串 s	98
4.4 本章小结	99
习题	99
第 5 章 函数	101
5.1 初识函数	101
5.1.1 函数的分类	101
5.1.2 函数的定义	103
5.1.3 案例分析：打印图案	105
5.2 函数的调用	106
5.2.1 函数调用的一般形式	106
5.2.2 函数的参数	108
5.2.3 函数的说明	109
5.2.4 案例分析：小型计算器	111
5.2.5 函数的嵌套调用	113
5.2.6 函数的递归调用	115
5.3 变量的作用域和存储域	118
5.3.1 变量的作用域	118

5.3.2 变量的存储类别	120
5.4 函数间的数据传递	124
5.4.1 形参和实参间的值传递	124
5.4.2 形参和实参间的地址传递	127
5.4.3 return 返回数据	128
5.4.4 全局变量传递数据	128
5.4.5 数组作参数	129
5.4.6 案例分析: 计算平均成绩	132
5.5 内部函数和外部函数	134
5.6 案例分析: 学生成绩管理程序	135
5.7 本章小结	138
习题	138

第 6 章 指针	141
6.1 指针是什么	141
6.2 指针变量	142
6.2.1 指针变量的定义	142
6.2.2 指针运算符	142
6.2.3 为何要使用指针	146
6.3 指针与数组	149
6.3.1 指向数组及数组元素的指针	149
6.3.2 指针变量的算术运算	151
6.3.3 案例分析: 输出数组全部元素	153
6.3.4 下标运算符[]的实质	154
6.4 指向多维数组的指针	155
6.4.1 使用二维数组名作为指针访问其元素	155
6.4.2 指向二维数组的指针变量	156
6.4.3 指针数组	160
6.4.4 指向指针的指针	160
6.4.5 案例分析: 输出二维数组全部元素	161
6.5 指针与字符串	164
6.5.1 字符串的表示方式	164
6.5.2 字符串的访问	165
6.5.3 字符串数组	167
6.6 函数型指针	169
6.7 指针型函数	170
6.8 动态分配内存	172
6.9 案例分析: 学生成绩管理程序	173

6.10 本章小结	182
习题	185
第7章 结构体与共用体	186
7.1 结构体类型定义和结构体变量说明	186
7.1.1 结构体类型变量的定义和引用	186
7.1.2 结构体类型变量的定义	188
7.1.3 结构体类型变量的引用	190
7.1.4 结构体类型变量的初始化	191
7.2 结构体数组的定义和引用	192
7.2.1 定义结构体数组	192
7.2.2 结构体数组的初始化	193
7.3 结构体指针的定义和引用	195
7.3.1 指向结构体类型变量的指针	195
7.3.2 指向结构体类型数组的指针的使用	196
7.3.3 案例分析:学生成绩管理程序(结构体指针)	198
7.4 链表	203
7.4.1 单链表结点类型的定义	204
7.4.2 单链表的建立	204
7.4.3 单链表的输出	206
7.5 共用体	208
7.5.1 共用体的定义	208
7.5.2 共用体变量的引用	210
7.6 枚举	211
7.6.1 枚举类型的定义和枚举变量的说明	211
7.6.2 枚举类型变量的赋值和使用	212
7.7 本章小结	213
习题	213
第8章 文件	215
8.1 文件的基本概念	215
8.1.1 文件概述	215
8.1.2 文件的类别	216
8.1.3 文件的操作流程	217
8.2 常用文件操作的标准函数	217
8.2.1 文件的打开	217
8.2.2 文件的关闭	218
8.2.3 文本文件的读写	219

8.2.4	二进制文件的读写	224
8.2.5	文件的其他常用函数	226
8.2.6	案例分析：文件操作	228
8.3	本章小结	230
	习题	230

第 9 章	综合实训 1	232
9.1	功能描述	232
9.2	程序主界面设计	232
9.3	功能项的详细设计	233
9.3.1	主界面函数的实现	234
9.3.2	初始化	237
9.3.3	数据录入	238
9.3.4	数据编辑	240
9.3.5	数据查询的实现	247
9.3.6	数据统计	251
9.3.7	数据导出的实现	253
9.3.8	数据导入	255
9.4	本章小结	256

第 10 章	综合实训 2	257
10.1	功能描述	257
10.2	程序主界面设计	257
10.3	功能项的详细设计	258
10.3.1	主界面的实现	259
10.3.2	初始化	259
10.3.3	数据录入	260
10.3.4	插入学生信息	263
10.3.5	信息的修改	266
10.3.6	信息的查询	268
10.3.7	信息的删除	270
10.3.8	显示学生信息	274
10.3.9	排序	275
10.3.10	数据回收	279
10.3.11	用户登录的实现	281
10.3.12	文件保存	284
10.3.13	文件读取	286

C 语言是继 BASIC 语言、FORTRAN 语言、COBOL 语言和 PASCAL 语言之后问世的一种通用计算机程序设计语言。早期的 C 语言主要用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识,到了 20 世纪 80 年代,C 语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛使用,成为最优秀的程序设计语言之一。它适用于编写各种系统软件、应用软件。

本章通过几个非常简单的例子来介绍 C 语言的结构特点、书写格式、输入输出函数以及如何用 Code::Blocks 实现 C 语言程序的运行等内容。

1.1 简单的 C 程序

1.1.1 一个简单的 C 程序

为了说明 C 语言源程序结构的特点,先看一个简单的例子。

【例 1.1】 原样输出一行语句

```
#include<stdio.h>           /* 输入输出函数编译预处理命令 */
void main()                 /* 主函数 */
{
    printf("Hello,world!\n"); /* 输出信息 */
}
```

程序运行结果如下:

```
Hello,world!
```

例 1.1 的功能是在屏幕上输出一行字符:

```
Hello,world!
```

程序是由文件组成的,include 称为文件包含命令,其意义是把尖括号(<>)或引号(" ")内指定的文件包含到本程序来,成为程序的一部分。被包含的文件通常是由系统提供的,其扩展名为.h,因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准

库函数的函数原型。凡是在程序中调用一个库函数时,都必须包含该函数原型所在的头文件。在本例中,使用了一个库函数:标准输出函数 printf。其头文件为 stdio.h 文件,因此在程序的主函数前用 include 命令包含了 stdio.h 文件。

main 是主函数的函数名,表示这是一个主函数。每一个 C 源程序都必须有且只有一个主函数(main 函数)。

printf 函数是一个由系统定义的标准函数,可在程序中直接调用。其功能是把要输出的内容显示到屏幕上。双引号内的\n 表示换行,在信息输出后,闪烁的光标将显示在屏幕的下一行。

程序中的“/* ... */”表示注释符号,在注释符号中间是注释内容,该内容可以由任何字符构成,系统不执行注释内容。注释的作用是为程序员阅读程序带来方便。

读者可尝试修改程序,在屏幕上显示自己感兴趣的字符或图形。

1.1.2 C 程序的结构特点

通过以上的分析,并结合下面的例子,可领会 C 语言程序的基本组成结构。

【例 1.2】 假设已知两个正整数 num1,num2,比较它们的大小,并输出其中的最大值。

```
#include<stdio.h>          /* 输入输出函数编译预处理命令 */
void main()                 /* 主函数 */
{
    int num1=3,num2=5,result; /* 定义变量 */
    result=max(num1,num2);    /* 调用自定义函数 max,并将结果赋给变量 result */
    printf("max=%d\n",result); /* 输出 result 的值 */
}

int max(int n1,int n2)       /* 定义函数 max,返回值为整型,n1,n2 为形式参数 */
{
    int r;                   /* 定义变量 */
    if(n1>n2)
        r=n1;                /* 如果 n1 比 n2 大,就把 n1 的值赋给 r */
    else
        r=n2;                /* 如果 n1 不比 n2 大,就把 n2 的值赋给 r */
    return r;                 /* 返回 r 的值,通过 max 返回到调用处 */
}
```

程序运行结果如下:

```
max=5
```

程序的功能是比较 num1 和 num2 的大小,并将其中的最大值输出在屏幕上。该程序由两个函数构成,一个是 main() 主函数,另一个是 max(n1,n2) 自定义函数(即用户自己设计的函数)。在主函数中既可以调用库函数(如 printf 函数),也可以调用自定义函数(如 max 函数)。

通过以上两个案例的分析,可以看出 C 语言有如下几个特点:

1. C 程序是由函数构成的

一个 C 源程序可由一个 main 函数和若干个其他函数组成,其中必须有且只有一个 main 函数。函数是 C 语言程序的基本单位。

2. main() 函数始终是 C 程序执行时的入口处

一个 C 语言程序总是从主函数 main() 开始执行,而不论 main() 函数在整个程序中的位置。

3. C 程序语句和数据定义必须以分号“;”结束

C 语言中,分号是程序语句的结束标志,也是 C 语句的必要组成部分。

4. C 语言严格区分大小写

在编写 C 语言程序时,一定要注意大小写,如将“printf”写成“Printf”,将会产生“'Printf' undefined”的警告,并发生连接错误。

一般 C 语言程序使用小写字母来书写程序,C 语言程序中的大写字母一般表示常量,请注意变量 a 和变量 A 系统认定为两个不同的变量。

5. C 语言用 /* 注释内容 */ 形式进行程序注释

在“/*”和“*/”之间的所有字符都为注释符,C 编译系统不对注释符进行编译。

1.1.3 C 程序的书写格式

C 程序书写格式非常自由,但从书写清晰、便于阅读理解、维护的角度出发,书写程序时应遵循以下规则。

1. 一个说明或一个语句占一行

C 语言中一行内可以写一条语句,也可以写多条语句。一条语句可以在一行内完成,也可以在多行内完成。但提倡一行一条语句的风格。

2. {} 的书写规范

用 {} 括起来的部分,通常表示了程序的某一层结构。一般情况下,左右大括号各占一行,并且需要上下对齐,这样便于检查大括号的成对性。在编写程序时,可先书写左右大括号,再编写括号中的内容,以避免括号不匹配的问题。

3. 程序书写采用缩进格式

根据语句的从属关系,程序书写时采用缩进格式,使程序语句的层次结构清晰,提高程序的可读性。同一层次语句要左对齐,低一层次的语句或说明可比高一层次的语句或说明缩进若干个字符,这样程序层次清楚,便于阅读和理解。

4. 程序中适当使用注释信息

编码过程中配合良好的注释,可增加程序的可读性、可维护性。

对于 C 语言程序的书写格式,读者可以从后面的程序中逐渐体会,编码时应力求遵循以上规则,以养成良好的编程风格。

1.2 C 语言概述

1.2.1 C 语言的产生及发展

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的,并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL(Basic Combind Programming Language) 发展演变而来。1970 年,AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 语言,最后导致了 C 语言的问世。

随着微型计算机的日益普及,出现了许多 C 语言版本,例如,Quick C、Microsoft C、Turbo C、C++、MS-C、Visual C 等。由于没有统一的标准,使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况,美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准,成为现行的 C 语言标准。

今天,越来越多的人在学习 C 语言,使用 C 语言,用 C 语言开发各个领域中的应用软件,C 语言已经是当今世界上最流行的程序设计语言之一。

1.2.2 C 语言的特点

C 语言发展如此迅速,而且成为最受欢迎的语言之一,主要因为它具有强大的功能。许多著名的系统软件,如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。下面就其主要特点简述如下。

1. C 语言简洁、紧凑

C 语言简洁、紧凑,使用方便、灵活。ANSI C 一共只有 32 个关键字、9 种控制语句,程序书写自由,主要用小写字母表示,压缩了一切不必要的成分。

2. C 语言集高级语言和低级语言的功能于一体

由于 C 语言实现了对硬件的编程操作,所以 C 语言具有直接访问硬件地址和寄存器的功能,因此具有较高的实时性。同时,也有高级语言面向用户、容易记忆、容易学习和书写的优点。C 语言集高级语言和低级语言的功能于一体,既可用于系统软件的开发,也适合于应用软件的开发。

3. C 语言是一种结构化语言

结构化语言的显著特点是代码及数据的分隔化,即程序的各个部分除了必要的信息交流外彼此独立。C 语言程序具有逻辑结构,有顺序、选择和循环 3 种基本结构,这种结构化方式可使程序层次清晰,便于使用、维护以及调试。

4. C 语言是便于模块化设计的语言

按模块化方式组织程序,有利于把整体程序分割成若干个相对独立的功能模块,便于团队开发。C 语言是通过函数来实现模块化设计的,这些函数为程序模块间的相互调用以及数据传递提供了方便。

5. C 语言具有较高的可移植性

C 语言是面向硬件和系统的,即与汇编语言比较接近,但它并不依存于机器硬件系统,便于在硬件不同的机种间实现程序的移植。因此广泛地移植到了各类各型计算机上,从而形成了多种版本的 C 语言。

1.3 C 语言程序的实现

1.3.1 运行 C 程序的步骤和方法

运行 C 程序的步骤如图 1-1 所示。

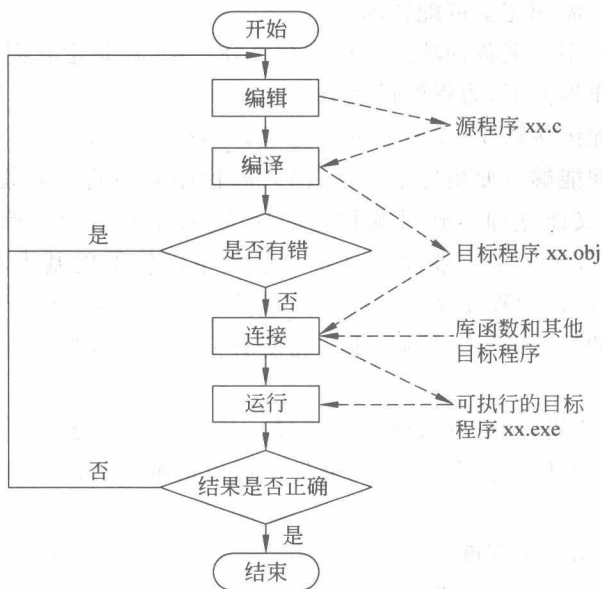


图 1-1 C 语言程序的运行过程

(1) 编辑: 将 C 语言源程序输入到编辑器中,并保存为文件,后缀名为“.c”。

(2) 编译: 将 C 语言源程序程序转变成机器语言程序。编译产生的程序称为目标程序,目标程序被自动保存为文件,这一文件称为目标文件,文件名的后缀是“.obj”。

本书采用 Code::Blocks 运行 C 语言程序。Code::Blocks 进行编译的依据是源程序,如果源程序中的符号、词语、整体结构等有差错,超出了 Code::Blocks 的“理解能力”,Code::Blocks 就无法完成编译,这样的差错称为语法错误。一旦发现语法错误,Code::Blocks 就不生成目标文件,并在窗口下方列出错误;如果没有语法错误,则生成目标文件,允许继续进行后面的步骤。

编译没有出现错误,仅仅说明程序中没有语法错误。

(3) 连接: 将目标程序和库函数或其他目标程序连接成 Windows 环境下的可执行文件,文件名后缀为“.exe”。

(4) 运行：运行可执行程序，观看输出结果是否正确。在运行这一步时，必须核对程序是否正确实现了预定的功能，如果功能不对，还必须到程序中寻找错误，纠正后再次经历(2)、(3)、(4)各步，直到看不出错误为止。运行时产生的错误称为逻辑错误，一般是由于算法错误或算法在转变为程序时出了问题，导致程序能够运行，却不能实现预想的功能。

1.3.2 Code::Blocks 集成开发环境的使用

集成开发环境(Integrated Developing Environment, IDE)是一个综合性的工具软件，它把程序设计全过程所需的各项功能集合在一起，为程序设计人员提供完整的服务。

Code::Blocks 是一个免费的 C、C++ 和 Fortran IDE 构建以满足最苛刻的用户的需求。它被设计成可扩展和完全可配置的。

集成开发环境并不是把各种功能简单地拼装在一起，而是把它们有机地结合起来，统一在一个图形化操作界面下，为程序设计人员提供尽可能高效、便利的服务。例如，程序设计过程中为了排除语法错误，需要反复进行编译、查错、修改、再编译的循环，集成开发环境就使各步骤之间能够方便快捷地切换，输入源程序后用简单的菜单命令或快捷键启动编译，出现错误后又能立即转到对源程序的修改，甚至直接把光标定位到出错的位置上。再如，集成开发环境的编辑器除了具备一般文本编辑器的基本功能外，还能根据 C 语言的语法规则，自动识别程序文本中的不同成分，并且用不同的颜色显示不同的成分，对使用者产生很好的提示效果。最后，IDE 和所有你需要的功能，有一个一致的外观、感觉和操作平台。

基于插件框架、代码块可以通过插件扩展。任何一种功能可以通过安装/编码添加插件。例如，编译和调试功能已经提供的插件。这也是本书选择 Code::Blocks 开发 C 语言程序的主要原因。

1. 启动 Code::Blocks 环境

方法：单击“开始”→“程序”→Code::Blocks 命令，启动 Code::Blocks，启动界面如图 1-2 所示。

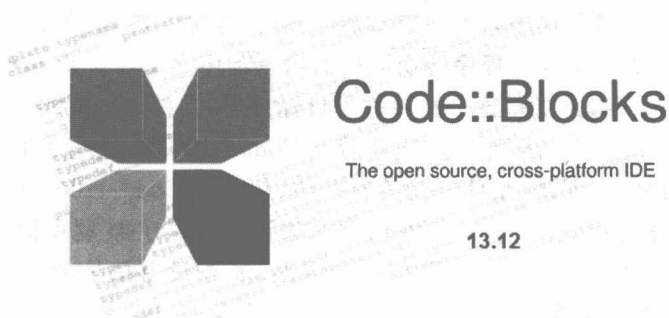


图 1-2 启动界面

初始界面如图 1-3 所示。创建成功的编译界面如图 1-4 所示。