

从零开始学

# C 程序设计

强调理论与实践相结合，按C的功能由浅入深地介绍程序设计语言的精要

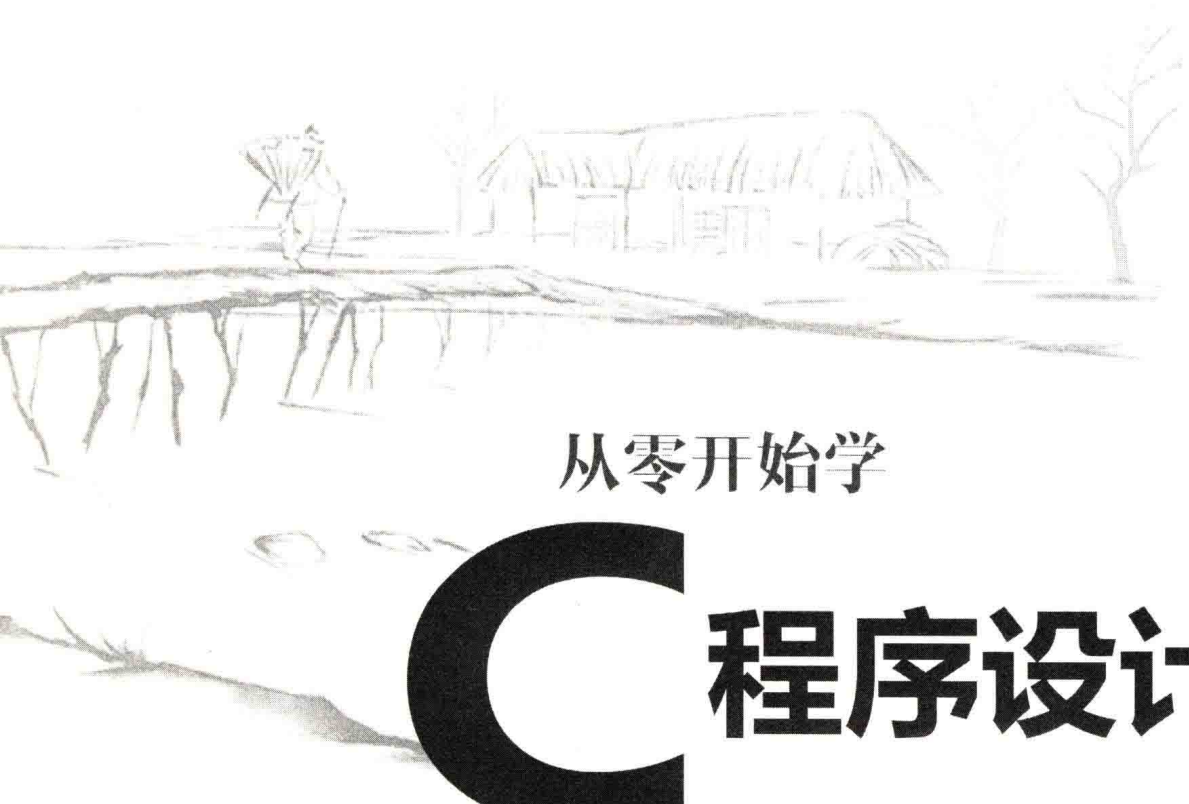
立即拥有结构化  
程序设计能力的 **16** 堂课

吴惠茹 等编著

大量范例程序，运行准确无误  
上机实训测验，强化编程与排错能力  
精心设计习题，验收教学成果



机械工业出版社  
China Machine Press



从零开始学

# C 程序设计

吴惠茹 等编著



机械工业出版社  
China Machine Press

## 图书在版编目 ( CIP ) 数据

从零开始学C程序设计 / 吴惠茹等编著. —北京: 机械工业出版社, 2017.5

ISBN 978-7-111-56470-6

I. ①从… II. ①吴… III. ①C语言-程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字 (2017) 第067123号

本书注重理论与实践结合, 按 C 语言的功能由浅入深地介绍 C 语言程序设计的精髓。

本书分 16 章说明 C 语言相关的语法与应用。首先介绍 C 语言的基础语法, 包括如何进行 C 语言程序的编写、编译、执行和除错, 通过对变量、常数、数据类型的学习, 进而了解各种运算符和流程控制指令; 然后介绍 C 语言的高级语法, 包括数组与字符串的声明与运用、指针的概念与实践, 并示范如何自定义函数、参数传递与函数的高级应用, 以及预处理器的使用; 最后介绍结构数据类型的基本概念、各种文件类型的操作技巧与管理以及从 C 到 C++ 面向对象程序设计的过渡。

本书的编写以教学为背景, 除了在各章正文的讲解中穿插大量范例程序的分析外, 在各章的后面还辅以课后习题与解答, 并提供丰富的上机程序测试题。本书适合作为大专院校计算机及相关专业的教材, 也适合作为程序设计初学者的自学教材, 同样可作为有一定编程经验、想快速掌握 C 语言的程序员的学习参考书。

## 从零开始学 C 程序设计

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 夏非彼 迟振春

印刷: 中国电影出版社印刷厂

版次: 2017 年 5 月第 1 版第 1 次印刷

开本: 188mm × 260mm 1/16

印张: 21.5

书号: ISBN 978-7-111-56470-6

定价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

# 前言

在计算机发展的几十年间，众多程序设计语言不断被各个时期的“达人”创造出来，进而不断被淘汰、取代、修订、融合或改头换面。和近年流行的程序设计语言（如 Python、C#、Java、C++、PHP、JavaScript 等）相比，C 语言显得有点“古董”，因为 C 语言的发展历史最悠久——设计思想萌芽于 1970 年年初，主体完成于 1973 年。但 C 语言在 2015 年仍然高居全世界所有程序设计语言使用人数的榜首，到 2016 年才被 Java 超过而屈居第二。

C 语言之所以长久不衰，是因为程序代码简洁高效、编译方式简易、能处理底层的存储器、产生的机器代码简短精悍，而且不需要复杂的系统运行环境便能高效运行。C 语言的这些特质深受广大程序员喜爱，并被广泛应用于操作系统和编译器的开发。例如，UNIX 和 Linux 就是基于 C 语言开发出来的，其他众多系统级的工具和各种高级程序设计语言的编译器或解释器大多也是使用 C 语言开发的。所以，C 语言被称为程序员的第一程序设计语言一点都不为过。

现在学 C 语言过时吗？作为一门通用计算机程序设计语言，C 语言远没有到过时的时候，只要学习 C 语言时不禁锢于面向过程程序设计思维，从零开始迅速掌握其精髓，而后补充面向对象程序设计的新思想，之后在学习 C++、C#、Java 或 Python 语言时就能得心应手。如果从一开始就学习上述 4 种面向对象的通用程序设计语言，会感到头疼不已。在出版本书的同时，我们还出版了一本《从零开始学 C++ 程序设计》供大家参考。C++ 语言 = C 语言 + 面向对象的概念，在 C#、Java 甚至 Python 中均可看到 C++ 的影子。

本书以教学为背景，分 16 章说明 C 语言相关的语法，除了在正文的讲解中穿插大量范例程序的分析外，还在各章末尾辅以课后习题与解答，并提供了丰富的上机程序测试题。本书适合作为大专院校计算机及相关专业的教材，也适合作为程序设计初学者的自学教材，同样可作作为有一定编程经验、想快速掌握 C 语言的程序员的学习参考书。

本书的范例程序有两类：一类是各章正文讲解使用的范例程序，另一类是各章后面“上机程序测验”提供的参考范例程序。读者可以从以下网址免费下载所有范例程序的源代码：

<http://pan.baidu.com/s/1nvDbllZ>（注意区分数字和字母大小写）

如果下载有问题，请发送电子邮件至 [booksaga@126.com](mailto:booksaga@126.com)，邮件主题设置为“求从零开始学 C 程序代码”。

全书所有范例程序都可以在标准 C 语言编程环境中编译通过和顺利运行。本书选用免费的 Dev C++ 5.11 集成开发环境对书中所有范例程序进行编译、修改、调试和测试，确保可以

准确无误地运行，读者可以放心参考、使用。另外，附录 A 包含“C 的标准函数库”，以便读者在学习的过程中速查常用的 C 语言标准函数的用法。附录 B 包含“C 编译程序的介绍与安装”，重点介绍 Dev C++集成开发环境的安装步骤和基本使用方法，读者可以在学习本书之前在自己的计算机上安装好 Dev C++集成开发环境。

本书主要由吴惠茹编著，卞诚君、王叶、周晓娟、刘雪连、吉媛媛、闫秀华、关静、孟宗斌、魏忠波、王翔、郭丹阳等人也参与了本书的编写与校对工作。虽然本书校稿过程力求无误，但是难免有疏漏之处，还望各位不吝赐教！

最后，祝大家学习顺利，迅速掌握 C 语言程序设计的精髓，进而成为使用 C 语言编程的高手，迈出成为合格程序员关键的一步。

编者

2017年2月

# 目 录

前言

第 1 章 C 语言的第一堂课.....	1
1-1 C 语言的起源.....	1
1-2 我的第一个 C 程序.....	2
1-2-1 程序代码编写规则.....	4
1-2-2 编译程序代码.....	5
1-2-3 运行 C 程序.....	5
1-2-4 程序代码的调试.....	6
1-3 helloworld 程序快速解析.....	7
1-3-1 头文件的作用.....	7
1-3-2 main()函数简介.....	8
1-3-3 system()函数的作用.....	9
1-3-4 注释与缩排.....	9
1-4 课后练习.....	10
第 2 章 变量与常数.....	13
2-1 认识变量.....	14
2-1-1 变量命名原则.....	14
2-1-2 变量的声明.....	15
2-2 变量的作用域.....	17
2-2-1 全局变量.....	18
2-2-2 局部变量.....	18
2-2-3 区块变量.....	18
2-3 常数.....	20
2-4 课后练习.....	22
第 3 章 C 语言的基本数据类型.....	24
3-1 认识基本数据类型.....	24
3-1-1 整数.....	24
3-1-2 浮点数.....	27

3-1-3	字符类型.....	29
3-1-4	转义字符简介.....	33
3-2	数据类型转换.....	34
3-2-1	自动类型转换.....	35
3-2-2	强制类型转换.....	36
3-3	上机程序测验.....	37
3-4	课后练习.....	38
<b>第4章</b>	<b>格式化输入与输出函数.....</b>	<b>41</b>
4-1	printf()函数.....	41
4-1-1	格式化字符.....	43
4-1-2	字段宽度设置功能.....	45
4-1-3	精度设置功能.....	47
4-1-4	标志设置功能.....	50
4-2	scanf()函数.....	51
4-2-1	格式化字符.....	52
4-2-2	加上提示字符.....	54
4-2-3	字段宽度设置功能.....	56
4-2-4	输入字符串.....	57
4-3	其他输入/输出函数.....	59
4-3-1	getchar()函数与 putchar()函数.....	59
4-3-2	getche()函数与 getch()函数.....	60
4-3-3	gets()函数与 puts()函数.....	62
4-4	上机程序测验.....	63
4-5	课后练习.....	64
<b>第5章</b>	<b>表达式与运算符.....</b>	<b>66</b>
5-1	表达式简介.....	66
5-2	认识运算符.....	67
5-2-1	赋值运算符.....	68
5-2-2	算术运算符.....	69
5-2-3	关系运算符.....	70
5-2-4	逻辑运算符.....	71
5-2-5	位运算符.....	74
5-2-6	递增与递减运算符.....	79
5-2-7	复合赋值运算符.....	81
5-3	上机程序测验.....	82

5-4	课后练习.....	82
<b>第 6 章</b>	<b>流程控制与选择性结构 .....</b>	<b>86</b>
6-1	顺序结构.....	86
6-2	选择结构.....	87
6-2-1	if 条件语句.....	88
6-2-2	if-else 条件语句 .....	90
6-2-3	条件运算符.....	94
6-2-4	switch 选择语句.....	95
6-3	上机程序测验.....	99
6-4	课后练习.....	100
<b>第 7 章</b>	<b>循环结构.....</b>	<b>103</b>
7-1	for 循环.....	104
7-1-1	for 循环的使用方式 .....	104
7-1-2	嵌套循环.....	106
7-2	while 循环.....	107
7-3	do while 循环.....	110
7-4	循环控制指令.....	112
7-4-1	break 指令.....	113
7-4-2	continue 指令.....	114
7-4-3	goto 指令.....	115
7-5	上机程序测验.....	117
7-6	课后练习.....	118
<b>第 8 章</b>	<b>数组与字符串.....</b>	<b>120</b>
8-1	认识 C 语言的数组 .....	120
8-1-1	一维数组.....	121
8-1-2	二维数组.....	126
8-1-3	多维数组.....	130
8-2	字符串简介.....	132
8-2-1	字符串的使用 .....	133
8-2-2	字符串数组.....	134
8-2-3	字符串处理功能.....	136
8-2-4	字符串处理函数.....	138
8-3	上机程序测验.....	140
8-4	课后练习.....	142



<b>第9章 指针基础入门</b> .....	<b>146</b>
9-1 认识地址.....	146
9-1-1 指针的作用.....	146
9-1-2 变量地址的存取.....	147
9-1-3 存取数组元素的地址.....	148
9-1-4 指针变量.....	149
9-2 多重指针.....	153
9-2-1 双重指针.....	153
9-2-2 多重指针.....	155
9-3 认识指针运算.....	157
9-3-1 递增与递减运算.....	157
9-3-2 指针常数与数组.....	159
9-3-3 指针变量与数组.....	162
9-4 上机程序测验.....	162
9-5 课后练习.....	163
<b>第10章 高级指针处理</b> .....	<b>166</b>
10-1 指针与字符串.....	166
10-1-1 使用指针设置字符串.....	166
10-1-2 指针数组.....	168
10-2 动态分配.....	170
10-2-1 动态分配变量.....	170
10-2-2 动态分配一维数组.....	172
10-2-3 动态分配字符串.....	174
10-2-4 动态分配多维数组.....	175
10-2-5 通用类型指针.....	180
10-3 上机程序测验.....	181
10-4 课后练习.....	182
<b>第11章 函数的基本认识</b> .....	<b>184</b>
11-1 认识函数.....	184
11-1-1 模块化设计精神.....	185
11-1-2 函数的使用.....	185
11-2 函数的使用.....	185
11-2-1 函数原型声明简介.....	185
11-2-2 函数的定义.....	187

11-2-3 函数的调用 .....	188
11-3 参数传递方式 .....	189
11-3-1 参数的意义 .....	189
11-3-2 传值调用 .....	189
11-3-3 传址调用 .....	192
11-3-4 数组参数的传递 .....	194
11-4 递归的作用 .....	197
11-5 上机程序测验 .....	199
11-6 课后练习 .....	201
<b>第 12 章 函数的高级应用与宏 .....</b>	<b>203</b>
12-1 命令行参数 .....	203
12-2 指针返回值 .....	205
12-3 函数指针 .....	207
12-4 变量的作用域 .....	208
12-4-1 auto 变量 .....	208
12-4-2 register 变量 .....	209
12-4-3 static 变量 .....	210
12-4-4 extern 变量 .....	211
12-5 预处理器 .....	213
12-5-1 宏指令 .....	214
12-5-2 条件编译指令 .....	216
12-5-3 #include 指令 .....	218
12-6 上机程序测验 .....	219
12-7 课后练习 .....	220
<b>第 13 章 结构数据类型 .....</b>	<b>223</b>
13-1 结构简介 .....	223
13-1-1 声明结构变量 .....	223
13-1-2 存取结构成员 .....	224
13-1-3 结构指针 .....	226
13-1-4 动态分配结构变量 .....	229
13-1-5 结构数组 .....	231
13-1-6 嵌套结构 .....	233
13-2 链表 .....	235
13-2-1 链表的建立 .....	236
13-2-2 链表程序的实现 .....	236

13-3	函数与结构.....	241
13-3-1	结构参数与传值调用.....	241
13-3-2	结构参数与传址调用.....	241
13-4	上机程序测验.....	242
13-5	课后练习.....	244
<b>第 14 章</b>	<b>其他自定义数据类型与项目设计.....</b>	<b>248</b>
14-1	类型定义指令.....	248
14-2	枚举指令.....	251
14-3	联合指令.....	253
14-4	项目程序简介.....	255
14-5	上机程序测验.....	259
14-6	课后练习.....	260
<b>第 15 章</b>	<b>文件的输入与输出.....</b>	<b>261</b>
15-1	缓冲区简介.....	261
15-1-1	fopen()函数与 fclose()函数.....	262
15-1-2	putc()函数与 getc()函数.....	264
15-1-3	fputs()函数与 fgets()函数.....	266
15-1-4	fprintf()函数与 fscanf()函数.....	268
15-1-5	fwrite()函数与 fread()函数.....	270
15-1-6	fseek 函数与 rewind()函数.....	273
15-2	无缓冲区的输入与输出.....	275
15-2-1	open()函数与 close() 函数.....	276
15-2-2	read()函数与 write() 函数.....	277
15-2-3	lseek()函数.....	280
15-3	上机程序测验.....	282
15-4	课后练习.....	283
<b>第 16 章</b>	<b>C 到 C++面向对象程序设计.....</b>	<b>286</b>
16-1	认识面向对象设计.....	286
16-1-1	封装.....	287
16-1-2	继承.....	288
16-1-3	多态.....	288
16-2	第一个 C++程序.....	288
16-2-1	头文件.....	289
16-2-2	程序注释.....	290

16-2-3	命名空间 .....	291
16-2-4	输入输出功能简介 .....	291
16-2-5	浮点数 .....	292
16-2-6	布尔数据类型 .....	294
16-3	C++的函数 .....	295
16-3-1	传引用调用 .....	296
16-3-2	内联函数 .....	297
16-3-3	函数重载 .....	299
16-4	类 .....	301
16-4-1	类的声明 .....	301
16-4-2	访问权限关键字 .....	302
16-4-3	创建类对象 .....	303
16-5	构造函数与析构函数 .....	305
16-5-1	构造函数 .....	305
16-5-2	析构函数 .....	309
16-5-3	作用域解析运算符 .....	310
16-6	上机程序测验 .....	313
16-7	课后练习 .....	314
附录 A	C 的标准函数库 .....	318
A-1	字符串处理函数 .....	318
A-2	字符处理函数 .....	319
A-3	常用数学函数 .....	320
A-4	时间与日期函数 .....	321
A-5	类型转换函数 .....	321
A-6	流程控制函数 .....	321
附录 B	C 编译程序的介绍与安装 .....	323
B-1	C/C++编译程序简介 .....	323
B-1-1	Visual Studio .....	323
B-1-2	Dev C++ .....	324
B-1-3	GCC .....	325
B-2	Dev C++的安装与介绍 .....	325
B-2-1	下载 Dev C++ .....	325
B-2-2	安装 Dev C++ .....	326
B-2-3	程序项目的建立 .....	328

# 第 1 章

## ◀ C语言的第一堂课 ▶

“程序设计语言”是人类用来和计算机沟通的语言，也是指挥计算机运行或运算的指令集合，可以将程序设计者的思考逻辑和语言转换成计算机能够理解的语言。C语言称得上是一种历史悠久的高级程序设计语言，也往往是初学者最先接触的程序设计语言，它对近代的程序设计领域有着非凡的贡献。

### 1-1 C语言的起源

1972年，贝尔实验室的Dennis Ritchie以B语言为基础，持续改善与发展，除了保留BCLP及B语言中的许多概念外，还加入了数据类型的概念和函数的功能，并且重新将它发表为C语言。在许多平台的主机上都有C语言的编译程序，例如MS-DOS、Windows系列操作系统、UNIX/Linux，甚至Apple公司的Mac系列系统等都有C语言的专用编译程序。许多程序设计人员使用C语言能够轻易地跨越不同平台来开发程序，因此让C语言广受科技界的欢迎。

由于各家厂商所开发和发布的C语言在编译程序时经常融入不同的特性与特殊的语法，因此给程序员在开发上增添了不少困扰。在20世纪80年代初（1980），美国国家标准局（American National Standard Institution）特别为C语言订制了一套完整的国际标准语法，称为ANSI C，最终成为了C语言的业界标准。于是1980年后，与C语言程序开发相关的工具一般都支持符合ANSI C的语法，所以大家在学习C语言时，使用最纯粹且符合ANSI C规范的C语言语法，几乎可以在各个平台上通行无阻。

经过数十年的发展，市场上众多厂商开发了许多种C语言编译程序，其中包含Borland公司的Turbo C/C++、Borland C++与Borland C++ Builder以及Microsoft公司的Visual C++等。另外，还有免费版本的C语言编译程序，包含MinGW、GCC、Dev C++等。

### C程序的特色

一般我们将程序设计语言分为高级语言与低级语言，但C语言经常被程序员称为中级语言，原因是C语言不但具有高级语言的亲和力（例如C语言的语法让人容易理解，可读性高，相当接近人类的习惯用语），而且在C语言的程序代码中允许开发者加入低级的汇编程序，使

得 C 程序能够与硬件系统直接沟通。

在还没有正式编写 C 程序之前，大家先要了解 C 本身属于一种编译式语言，也就是使用编译程序（Compiler，或称为编译器）将源程序转换为机器可读的 executable 文件或目标程序，不过编译程序必须先把源程序读入主存储器（内存）才可以开始编译。

编译后的目标程序可直接对应成机器码，故可在计算机上直接执行，不需要每次运行时再重新编译，执行速度自然较快。每次修改源程序，必须重新经过编译程序编译，才能保持运行文件为最新版本。

### 提示

解释式语言与编译式语言不同，前者是使用解释器（Interpreter）来对高级语言的源代码进行逐行解释，每次解释完一程序语句才会解释下一程序语句。解释的过程中如果发生错误，解释动作就会立刻停止。由于使用解释器解释的程序每次运行时都必须再解释一次，因此运行速度较慢，不过因为只需存取源程序、不需要转换为其他类型的文件，所以占用的内存较少。例如，BASIC、Lisp、Prolog 等语言都使用解释执行的方法。

C 语言可以直接处理底层内存，甚至用于实现位逻辑运算，因此所能实现的功能不仅仅限于开发常规的软件包，还可以开发硬件驱动程序、网络协议以及嵌入式系统等。特别值得一提的是以 C 语言开发出来的程序文件容量相对较小，并且不需要依赖虚拟机或运行时（runtime）环境就可以直接运行。与 Java、Visual Basic、Pascal 等程序设计语言相比，C 语言的执行效率相当高，运行时也相当稳定。例如，相当知名的开放源码操作系统——Linux 便是以 C 语言所编写而成的，一般对修改 Linux 源代码有兴趣的学者与工程师，肯定都要有 C 语言的基础才能够入门。C 的设计模式与语法深深影响了许多后来发展出来的程序设计语言，最显著的例子是 C++、Java、C# 等。

## 1-2 我的第一个C程序

其实学习程序设计语言和学游泳一样，下水直接体验才是最快的方法。从笔者多年从事程序设计语言的教学经验中得出这样的结论：在教初学者学习新的程序设计语言时，废话不要太多，让他们从无到有、实际编写和运行一个程序最为重要，许多编程高手都是程序写多了，对所使用的程序设计语言的领悟就越来越深。

早期，要学习 C 语言程序设计，首先必须找一种文本编辑器来进行程序的编辑，例如 Windows 系统下的“记事本”编辑器，或者 Linux 系统下的 vi 编辑器，接着选一种 C 语言的编译程序（如 Turbo C/C++、MinGW、GCC 等）编译，然后运行。不过现在不用这么麻烦了，只要找个可将程序的编辑、编译、运行与调试等功能集成于同一个操作环境下的“集成开发环境”（Integrated Development Environment，IDE）即可。

由于 C 语言的应用市场很大，市面上较为知名的 IDE 就有 Dev C++、Visual C++ Express、C++ Builder、Visual C++ 和 GCC 等。现在流行的几种 C/C++ 集成开发环境都有一些自定义的



语法与特殊功能。对于初学者而言，只要从基本的内容着手，将重点放在语法、逻辑等方面就可以了。目前市面上几乎没有单纯的 C 语言编译程序，通常都是与 C++ 编译程序兼容，称为 C/C++ 编译程序或编译器。注意，本书中所有的 C 程序文件都是以免费的 Dev C++ 集成开发环境来进行编译与运行的。

现在请各位读者按照附录 B 的说明，在你的计算机中安装好 Dev C++，然后开始运行 Dev C++ 集成开发环境，随后就会出现程序运行界面，如图 1-1 所示。

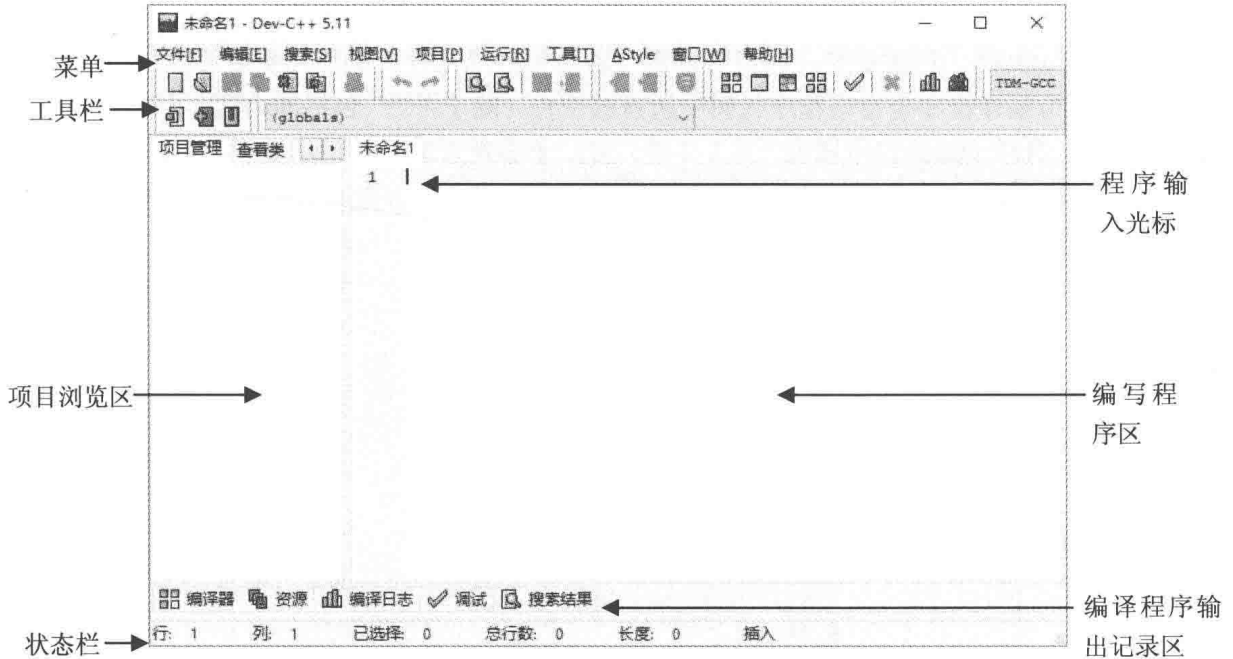


图 1-1 Dev C++ 集成开发环境

接下来带领大家使用 Dev C++ 来编写第一个程序 helloworld (文件名)。首先要打开的是单个文件的功能，也就是编写单个程序。请选择“文件”→“新建”→“源代码”以打开一个新的源代码文件，然后在 Dev C++ 的程序代码编辑区中输入第一个 C 语言练习程序代码：

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    int no;
    no=2;
    printf("There are %d pandas in Beijing.\n",no);
    /*输出北京有两只熊猫*/

    system("PAUSE");
    return 0;
}
```

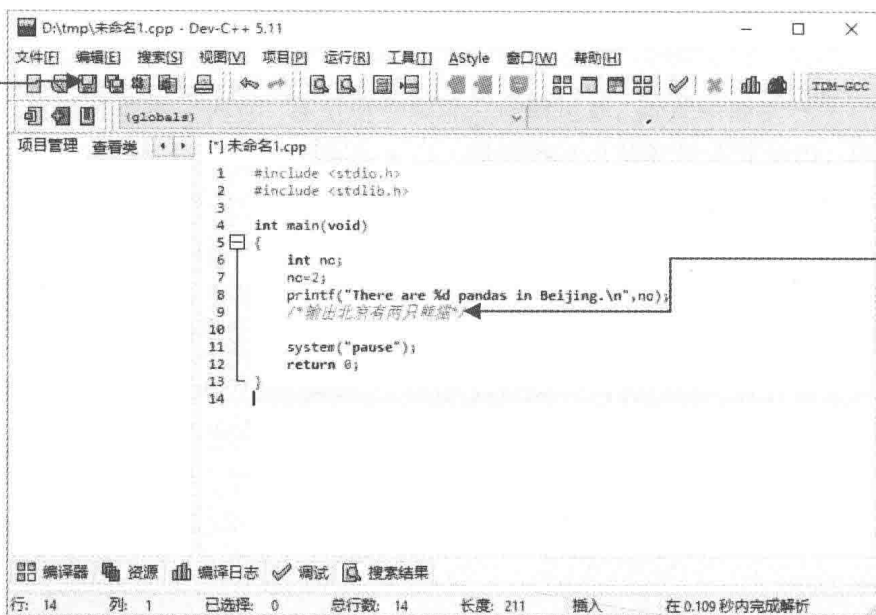
## 1-2-1 程序代码编写规则

当我们开始在 Dev C++ 中输入程序代码时，C 语言的程序代码编写采用的是自由格式 (free format)，也就是只要不违背基本语法规则，就可以自由安排程序代码的位置，不过字母大小写是有区分的。请注意，Dev C++ 是一种可视化的窗口编辑环境，而且还会将程序代码中的字符串 (红色)、指令 (黑色) 与注释 (深蓝色) 标示成不同颜色。

每一条语句以 “;” (分号) 作为结尾与分隔，中间有空格符、Tab 键、换行，它们都算是空白 (white space)。也就是说，我们可以将一条语句拆成几行，或将几条语句放在同一行。

这是因为编译程序会忽略程序代码中所有的空白 (除了 “” 包括的内容，因为它属于字符串内容)，只有当编译程序遇到 “;” (分号) 时，才会判定该条语句结束。在同一条语句中，完整不可分割的单元称为标记符号 (token)，两个标记符号间必须以空格键、tab 键或回车键来分隔，如图 1-2 所示。

2. 程序写完后，单击“保存”按钮，并确定存盘路径、文件名 helloworld，以.c 作为扩展名



1. 请自行输入 C 程序代码

图 1-2 C 语言程序在 Dev C++ 集成开发环境中显示的样子

如果这个文件是新建的文件，而且尚未存盘，Dev C++ 会提醒你先将该文件存盘。在此我们将文件存为 helloworld.c，如图 1-3 所示。



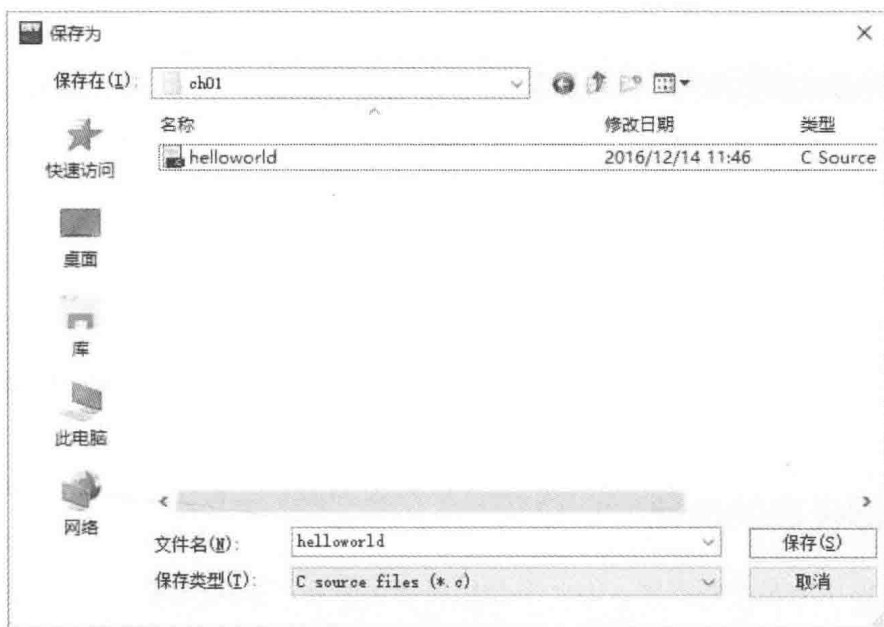



图 1-3 将新建的 C 语言程序文件存盘

## 1-2-2 编译程序代码

接下来开始执行编译过程，单击工具栏中的“编译”按钮或依次选择“运行”→“编译”菜单选项，接着在“编译日志”窗格中显示编译过程，代表文件正在编译，如果编译成功，就会显示出如图 1-4 所示的编译结果。

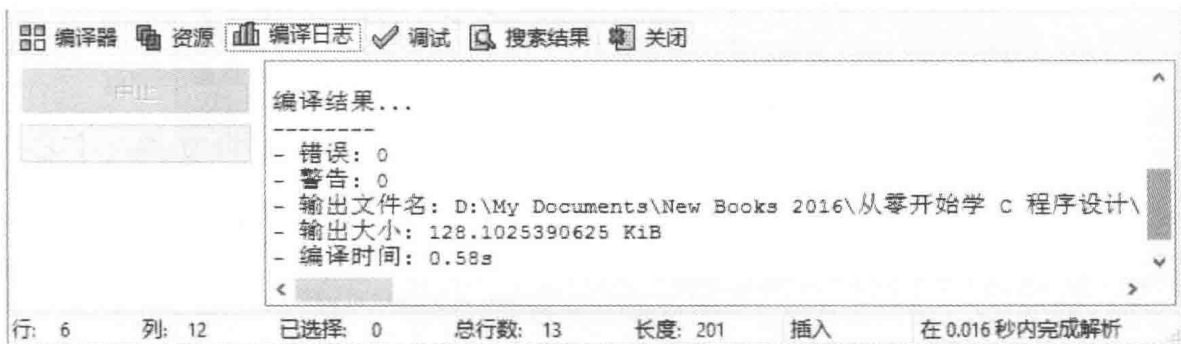


图 1-4 编译成功后显示的编译日志

编译阶段包括“编译”“链接”两个步骤，如果没有语法错误，编译程序就会把翻译结果存成目标文件（object file）。目标代码是一种二进制文件，此文件的扩展名为“\*.obj”，这个目标文件经由链接程序（linker）链接到其他目标文件和函数库后，最后生成可执行文件。由于在 Dev C++中默认使用完这个目标文件后会删除，因此一般看不到这类文件。

## 1-2-3 运行 C 程序

可执行文件的扩展文件名在 Windows 系统下是“\*.exe”，当 C 语言的程序代码“摇身一