

高等学校计算机基础教育规划教材

# 程序设计基础

罗 兵 高 潮 洪智勇 编著

清华大学出版社





## 内 容 简 介

以 C 语言结合 C++ 的输入输出流技术对工科学生进行编程基础教学,包括传统经典的 C 语言语法、面向过程的模块化程序结构方法,为进一步进行单片机、嵌入式等应用程序开发打下基础。本书以成果导向的教学(Outcome Based Education,OBE)理念进行内容组织,侧重于对学生编程能力的培养,注重应用,淡化语法细节,多案例、细讲解、少语法、少偏怪。目标是使学生了解必要的语法,具有程序设计的能力。本书采用 C 语言作为编程基础语言,同时利用 C++ 兼容 C 的特点,用 C++ 程序进行编程示例,输入输出采用 C++ 的输入输出流技术,其余部分均采用 C 语言的语法。本书有配套的实验指导《程序设计实验指导书》,还配有教学课件、例题和习题源程序等电子资源。

本书适用于应用型高校计算机类、信息类、控制类专业学生的程序设计学习。

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/罗兵,高潮,洪智勇编著. —北京:清华大学出版社,2019

(高等学校计算机基础教育规划教材)

ISBN 978-7-302-53262-0

I. ①程… II. ①罗… ②高… ③洪… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 133423 号

责任编辑:袁勤勇 杨 枫

封面设计:常雪影

责任校对:梁 毅

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市国英印务有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:17.5

字 数:426 千字

版 次:2019 年 9 月第 1 版

印 次:2019 年 9 月第 1 次印刷

定 价:39.00 元

产品编号:083932-01

# 前言

现代科技的发展离不开计算机,现代工科大学生大多需要掌握计算机编程技术,因此程序设计课通常是各高校学习人数多、开设专业多的一门必修课。对于应用型本科学生来说,既感受到计算机编程的重要作用,但又常被复杂烦琐的语法所困扰,被应试教育所累。现代工程教育提出了成果导向的教育理念(Outcome Based Education, OBE),注重应用能力和自学能力的培养,以案例为导向,淡化知识的系统教学,系统知识可以留待自学和遇到需要时再学习。这样会提高学习兴趣,使学习目的更明确,能更好地将理论结合实际,学以致用。

传统的程序设计教学内容偏重语法体系,注重偏、难、怪的语法细节,程序例题和习题往往脱离实际应用,容易导致应试导向,而使学生对学习目的感觉迷茫,挫伤了学生的学习热情,课程学完后往往只会应试,不会编程应用。

本书将现代工程教育提出的成果导向的教育理念应用于高校工科程序设计课教学改革实践,内容上突出重点,注重实例,淡化意义不大的偏、难、怪的语法细节问题。

目前,程序设计一般采用的C语言有诸多适合做编程入门语言的特点,如面向过程、结构化程序设计、规范、清晰、功能强、可直接控制底层、可直接访问硬件、与多种语言有相似性、容易再学习新的编程语言等。但C语言的标准输入输出函数使用很不方便,往往需要初学者花费很多时间和精力进行学习和程序调试。为此,本书采用C语言作为编程基础语言,同时利用C++兼容C的特点,用C++程序进行编程示例,输入输出采用C++的输入输出流技术,其余部分均采用C语言的语法,使学生更多地关注于程序结构和算法,提高程序开发兴趣和能力,学以致用。

全书共有12章,分别介绍了C语言的基础知识、数据类型、运算符、表达式、程序结构、函数、数组、指针、字符串、构造数据类型、文件等内容,最后一章是程序设计的综合应用。本书还有6个附录,列举了ASCII码表、C语言的关键字、运算符的优先级、库函数等内容。

本书适合作为应用型高校工科专业学生学习程序设计课程的教材,也可作为程序设计初学者的入门教材。本书配套有《程序设计实验指导书》,还配有教学课件、例题和习题源程序等电子资源。

本书由罗兵、高潮、洪智勇编著，罗兵编写了第1、2、3、9、10、11章，高潮编写了第4~8章，洪智勇编写了第12章并负责全书的统稿工作。

由于编者知识水平有限加之时间紧迫，诸事繁忙，书中难免存在不少错误和不足，恳请广大读者指正。

编者

2019年6月

# 目 录

<b>第 1 章 基础知识</b> .....	<b>1</b>
1.1 为什么要学习程序设计 .....	1
1.2 C 语言简介 .....	3
1.3 相关的软件知识 .....	8
1.3.1 二进制.....	8
1.3.2 源程序到可执行程序.....	9
1.3.3 预处理命令 .....	10
1.3.4 输入输出 .....	11
1.4 相关的硬件知识.....	11
1.4.1 计算机的硬件体系结构 .....	11
1.4.2 内存的结构原理 .....	12
1.5 小结.....	14
习题 1 .....	14
<b>第 2 章 数据类型</b> .....	<b>16</b>
2.1 数据类型的概念.....	16
2.2 常量与变量.....	17
2.2.1 常量 .....	18
2.2.2 变量 .....	19
2.2.3 标识符的命名规则 .....	20
2.3 整型数据.....	20
2.3.1 无符号整型数据 .....	21
2.3.2 有符号整型数据 .....	21
2.4 实型数据.....	23
2.4.1 实型常量 .....	23
2.4.2 实型数据的存储格式 .....	23
2.5 字符型数据.....	24
2.6 字符串常量.....	25



2.7	不同类型数据的混合运算	26
2.8	不同类型数据的输入输出	29
2.8.1	C++ 的输入输出	29
2.8.2	C 语言的格式输入输出函数	30
2.9	小结	38
	习题 2	38
<b>第 3 章</b>	<b>运算符及表达式</b>	<b>40</b>
3.1	基本概念	40
3.1.1	运算符分类	41
3.1.2	运算符与数据类型	41
3.1.3	运算符的优先级与结合性	42
3.2	算术运算符	42
3.3	赋值运算符	43
3.4	关系运算符	45
3.5	逻辑运算符	46
3.6	位运算符	47
3.6.1	位逻辑运算符	47
3.6.2	移位运算符	48
3.6.3	位运算的应用	49
3.7	运算符的优先级	49
3.8	小结	50
	习题 3	50
<b>第 4 章</b>	<b>程序的选择结构</b>	<b>52</b>
4.1	双分支选择结构	53
4.2	单分支选择结构	57
4.3	选择结构语句的嵌套	60
4.4	多分支选择结构	62
4.4.1	级联式 else if 语句	62
4.4.2	switch 语句	64
4.5	小结	68
	习题 4	68
<b>第 5 章</b>	<b>程序的循环结构</b>	<b>70</b>
5.1	程序的循环控制	70
5.2	while 语句	73
5.3	do-while 语句	77

5.4	for 语句	79
5.5	循环结构的嵌套	86
5.6	特殊的流程控制语句	90
5.6.1	break 语句	90
5.6.2	continue 语句	90
5.6.3	goto 语句	92
5.6.4	exit()函数	93
5.7	穷举算法与迭代算法	93
5.7.1	穷举算法	94
5.7.2	迭代算法	95
5.8	小结	96
	习题 5	96
<b>第 6 章</b>	<b>函数</b>	<b>100</b>
6.1	函数的定义与调用	101
6.1.1	函数定义	101
6.1.2	函数调用	103
6.2	如何建立函数	106
6.2.1	建立函数的基本方法	106
6.2.2	函数封装与程序的健壮性	108
6.3	函数原型与函数声明	111
6.4	递归函数	118
6.5	变量的作用域与存储类型	122
6.5.1	局部变量与全局变量	122
6.5.2	变量的存储类型	125
6.6	模块化程序设计	127
6.7	小结	129
	习题 6	129
<b>第 7 章</b>	<b>数组</b>	<b>135</b>
7.1	数组的定义与初始化	136
7.1.1	数组的定义	136
7.1.2	数组的初始化	137
7.2	数组的引用	139
7.3	数组作函数参数	142
7.4	排序问题	147
7.4.1	交换排序法	147
7.4.2	选择排序法	148

7.4.3	冒泡排序法	150
7.5	查找问题	153
7.5.1	顺序查找	154
7.5.2	折半查找	155
7.6	字符数组	158
7.7	小结	162
	习题 7	162
<b>第 8 章 指针</b>		<b>166</b>
8.1	指针概述	167
8.1.1	指针与地址	167
8.1.2	指针变量的定义与指针运算符	168
8.1.3	指针作函数参数	172
8.2	指针与一维数组	174
8.2.1	指向一维数组元素的指针	175
8.2.2	数组名和指针作函数参数	177
8.3	指针与二维数组	179
8.3.1	指向指针的指针	179
8.3.2	二维数组的指针	179
8.3.3	指向二维数组的指针变量	182
8.3.4	二维数组指针作函数参数	183
8.4	指针与字符串	186
8.5	函数的返回值为指针	188
8.6	动态内存分配	191
8.6.1	malloc 函数	191
8.6.2	calloc 函数	192
8.6.3	free 函数	193
8.7	小结	194
	习题 8	194
<b>第 9 章 字符串</b>		<b>200</b>
9.1	字符串常量	200
9.2	字符串处理函数	201
9.3	字符串与字符数组	205
9.4	小结	205
	习题 9	206

<b>第 10 章 构造数据类型</b> .....	<b>207</b>
10.1 结构体类型 .....	207
10.1.1 结构体类型的声明 .....	207
10.1.2 结构体变量的定义 .....	208
10.1.3 结构体变量成员的引用 .....	209
10.2 共用体类型 .....	211
10.3 枚举类型 .....	213
10.4 链表 .....	216
10.4.1 创建链表 .....	216
10.4.2 对链表的基本操作 .....	217
10.5 小结 .....	220
习题 10 .....	221
<b>第 11 章 文件的操作</b> .....	<b>222</b>
11.1 文件类型及打开方式 .....	222
11.2 读文件 .....	223
11.3 写文件 .....	224
11.4 小结 .....	231
习题 11 .....	231
<b>第 12 章 综合应用</b> .....	<b>232</b>
12.1 典型程序示例 .....	232
12.2 程序设计实例 .....	244
12.3 单片机程序实例 .....	250
12.4 小结 .....	252
习题 12 .....	252
<b>附录 A ASCII 码表</b> .....	<b>253</b>
<b>附录 B C 语言的关键字</b> .....	<b>256</b>
<b>附录 C C 语言运算符的优先级和结合性</b> .....	<b>257</b>
<b>附录 D C 语言的常用库函数</b> .....	<b>259</b>
<b>附录 E C 语言语法概要</b> .....	<b>264</b>
<b>参考文献</b> .....	<b>269</b>

## 基础知识

## 1.1 为什么要学习程序设计

现代科学技术的发展离不开计算机,从智能手机到飞机、高铁,从工业自动控制到人脸识别,无处不见计算机的身影。计算机包括硬件和软件两部分,硬件的核心是中央处理器(CPU);软件的核心是程序(Program)。

无论从事电子、通信、自动化、电气、机械、建筑、化工、纺织、材料、能源、交通、设计等哪一个领域的工作,或者就是从事计算机行业本身,都需要掌握程序设计技能,才能更好地使用计算机,使其成为专业技术工作的利器。

本书采用C语言作为工科程序设计的入门语言。

下面通过几个例子,看看C语言程序可以做什么。

**例 1.1** 将一张 100 元的纸币换成 10 元、20 元和 50 元的零钞,有几种换法?

**【解】** 设将一张 100 元纸币换成  $x$  张 10 元、 $y$  张 20 元和  $z$  张 50 元零钞,这是一个求解三元一次方程的整数解问题:  $10x+20y+50z=100$ 。

计算机求解此类问题,可以采用穷举法,因为计算机擅长做有规律的重复计算工作。

将  $x$  从 0 到 10 循环、 $y$  从 0 到 5 循环、 $z$  从 0 到 2 循环,每次将循环的候选解代入方程的左边计算判断结果是否为 100,为 100 则输出为一组解。该问题的计算机程序为:

```
#include<iostream>
using namespace std;
int main(void)
{
    int x,y,z;
    for(x=0;x<=10;x++)
        for(y=0;y<=5;y++)
            for(z=0;z<=2;z++)
                if(10*x+20*y+50*z==100)
                    cout<<x<<" * 10+"<<y<<" * 20+"<<z<<" * 50=100"<<endl;
    return 0;
}
```

程序运行结果如图 1-1 所示。

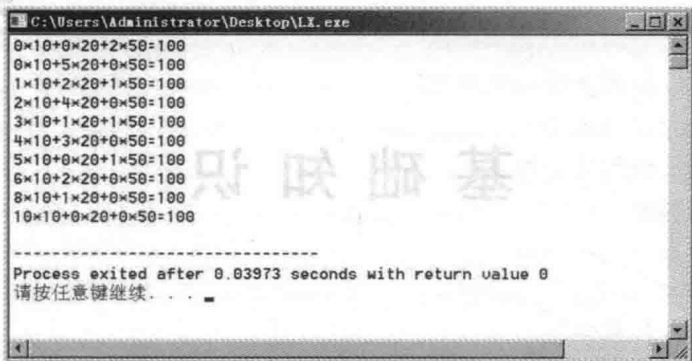


图 1-1 例 1.1 的程序运行结果

例 1.2 求解关于  $x$  的一元二次方程  $ax^2+bx+c=0$ 。

【解】 C 语言程序的算法如图 1-2 所示。

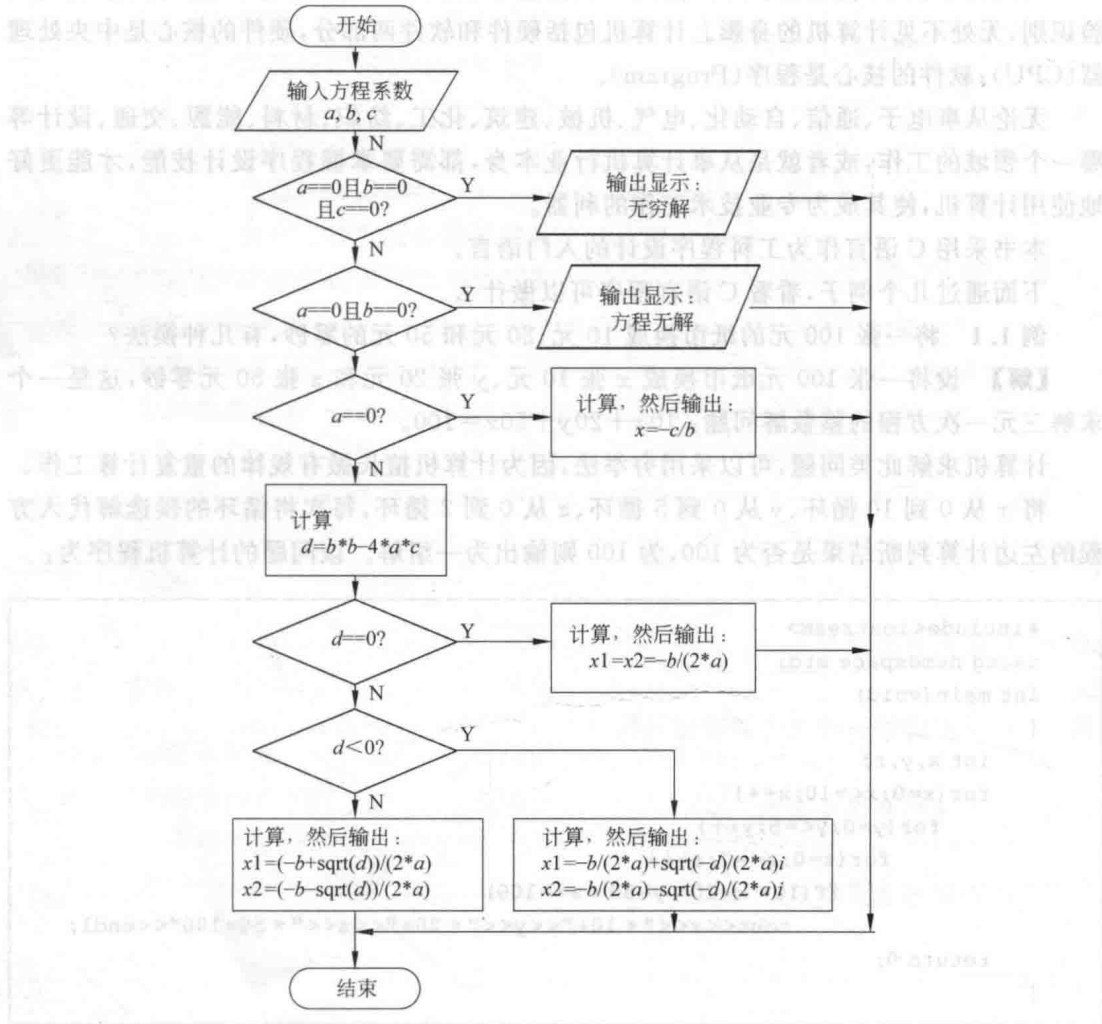


图 1-2 求解一元二次方程的算法流程图

计算机程序擅长解决复杂的计算问题或复杂的逻辑问题,其他复杂问题只要能够将要求解的问题转换为计算问题和逻辑判断问题,计算机程序都可以解决。“记忆好(存储量大)”、“反应快(计算快、存储快)”、擅长做重复有规律的处理(循环)是计算机程序的特点。

本课程首先学习如何将实际问题用计算机 C 语言描述,并求解。从使用的角度来说,C 语言还是比较简单的,大家要有信心掌握。当然,C 语言的语法细节又有很多比较困难的地方,但对于实际应用意义不大。例如,变量  $a$  的值原来为 1,分析执行语句“ $a+=a-=a*=a;$ ”后变量  $a$  的值。这种语句是实际意义不大的。真正的难点是在于如何将复杂的问题转换为计算机可解决的计算模型,即算法问题。例如,如何求圆周率?在计算机只能直接进行加减乘除四则算术运算的条件下如何计算平方根?

本课程主要学习将已设计好的算法写为 C 语言程序,即算法实现。

## 1.2 C 语言简介

在 C 语言诞生以前,系统软件主要是用汇编语言编写的。由于汇编语言依赖于计算机硬件,与计算机的具体型号密切相关,其可读性和可移植性都很差;但一般的高级语言又难以实现对计算机硬件的直接操作,于是产生一种兼有汇编语言和高级语言特性的新语言的强烈需求。20 世纪 70 年代美国最早设计出现了 C 语言。

美国电报电话公司(AT&T)的贝尔实验室工程师 Ken Thompson 在 1970 年以 BCPL(Basic Combined Programming Language)语言为基础,设计了一种类似于 BCPL 的语言,取 BCPL 的第一个字母 B,称为 B 语言。1972 年贝尔实验室的 Dennis M. Ritchie 为克服 B 语言的诸多不足,在 B 语言的基础上重新设计了一种语言,取 BCPL 的第二字母 C,故称为 C 语言。BCPL 和 B 语言最初都是为编写操作系统软件和编译器而开发的语言。贝尔实验室在 B 语言的基础上开发出的 C 语言,最初也是用来编写 UNIX 操作系统的。C 语言是作为 UNIX 操作系统的开发语言为人们所认识,但由于 C 语言严格的设计,其与具体硬件无关以及其他许多优点,使它的应用很快就超越了贝尔实验室的范围。20 世纪 70 年代末,C 语言开始移植到非 UNIX 环境中,并逐步脱离 UNIX 系统成为一种独立的程序设计语言,迅速地在全球传播。直到现在,C 语言始终在全球编程语言使用率排名中稳居前三名。图 1-3 是国际著名的软件测评公司 TIOBE 最新的编程语言使用率排名。

C 语言之所以备受青睐,是和它具有的许多优点分不开的,这些优点具体如下。

(1) C 作为一种高级语言,又具有机器语言的功能。高级语言与人的语言习惯、表达方式更接近,容易理解、掌握和使用,包括数学表达式。C 语言又能直接访问物理地址、具有位(bit)操作的功能,可以直接对硬件进行操作。这使得 C 语言成为单片机、小型智能设备等的程序设计语言,并一直是使用率最高的程序设计语言之一。

(2) 用 C 语言编写的程序编译后所生成的目标代码质量高、程序的执行效率高。由于可直接访问控制硬件、代码效率又高、又具有高级语言的特点,所以在追求速度的小型

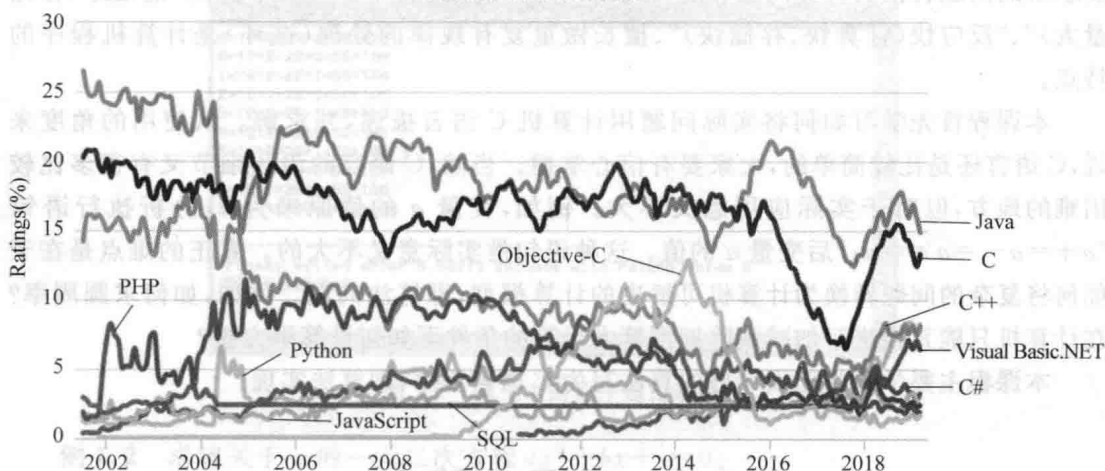


图 1-3 TIOBE 公司最新发布的编程语言使用率排名

智能设备、工业控制和实时处理中被广泛采用。

(3) C 语言是典型的面向过程的结构化程序设计语言,具有结构化的控制语句。顺序、选择、循环这 3 种类型的结构化控制,在 C 语言中都有相应的语句加以体现,因此可以用 C 语言进行结构化程序设计。C 语言还以函数作为程序的基本单位,便于实现程序的模块化。面向对象程序设计或其他高级语言都需要类似的面向过程的程序设计逻辑基础。所以 C 语言也一直被选为程序设计入门教学语言。

(4) C 语言简洁紧凑,功能强大。它只有 32 个关键字和 9 种控制语句,而且书写形式自由,所以 C 语言入门比较容易。它又有丰富的数据类型和丰富的运算符,可以实现强大的运算和逻辑功能。它不仅有常用的基本数据类型,而且还有指针类型、构造数据类型等,能用来实现各种复杂的数据结构。C 语言的丰富运算符使得许多操作都可以通过运算符来表示。由运算符和操作数再组成表达式,因而 C 语言中表达式的类型也是相当丰富的。

因为以上这些优点,C 语言既可以用来开发系统软件,也可以用来开发应用软件。另一方面,由于 C 语言的语法限制不太严格,程序设计自由度大,对程序员要求也较高。而且 C 语言是一种面向过程的程序设计语言,面向过程的程序设计是以过程处理为核心,这种设计方法在大型程序设计、软件的重用性、图形化界面设计等方面有其局限性。

20 世纪 80 年代初以来,随着面向对象程序设计思想的日益普及,很多支持面向对象程序设计方法的语言也相继出现,C++ 就是其中之一。C++ 是 Bjarne Stroustrup 于 1980 年在 AT&T 的贝尔实验室开发的一种语言。它是 C 语言的超集和扩展,是在 C 语言的基础上扩充了面向对象的语言成分而形成的。最初这种扩展后的语言称为带类(class)的 C 语言,1983 年才被正式称为 C++ 语言。以后又经过不断的改进,发展成为今天的 C++。

作为 C 语言的超集,C++ 继承了 C 的所有优点。它既保持了 C 的简洁性和高效性,

又对数据类型做了扩充,使得编译系统可以检查出更多类型错误。C++ 支持面向对象程序设计,通过类和对象的概念把数据和对数据的操作封装在一起,通过抽象、封装、继承和多态等特性实现了软件重用和程序自动生成,使得大型复杂软件的构造和维护变得更加有效和容易。

C++ 完全兼容 C,很多用 C 编写的库函数和应用程序都可以为 C++ 所用。正是由于与 C 兼容,使得 C++ 不是纯粹的面向对象的语言,它是一种既支持面向对象也支持面向过程的程序设计语言。

由于 C 语言的广泛应用,C++ 语言又对 C 语言具有向下兼容性,以及对面向对象技术的支持,使其在推出后很快就获得商业上的成功。各大公司纷纷推出自己的 C++ 语言编译器和开发工具,其中微软公司的 Visual C++ 是比较流行的 C++ 语言开发平台之一。Dev-C++ 是一个 Windows 环境下的适合于初学者使用的轻量级 C/C++ 集成开发环境 (IDE)。它是一款免费软件,要求在遵守通用公共协议 (GPL) 前提下复制、使用。它集合了 MinGW 中的 GCC 编译器、GDB 调试器和 AStyle 格式整理器等众多免费软件。原开发公司 Bloodshed 在开发完 4.9.9.2 后停止开发,现在由 Orwell 公司继续更新维护,最新版本为 5.9.2。本书采用 Dev-C++ 作为开发平台。

由于 C 语言的输入输出语法复杂而且使用意义不大,而 C++ 中的输入输出流简洁方便,所以本书的所有例子均采用 C++ 的输入输出流,而其他语法仍然采用 C 语言的语法,而 C++ 可以兼容这些语法。所以本书几乎所有的例子程序文件后缀为 cpp 而不是 c。

通过本书程序设计基础的学习,读者可以掌握以下几种技能。

- (1) 面向过程的 C 语言程序设计,能编写结构化程序;
- (2) 掌握计算机的程序逻辑,掌握分支、循环、穷举等简单算法,将用户需求转化为计算机语言实现;
- (3) 为后续的单片、智能控制设备软件开发打下基础,也为计算机专业学生学习其他高级语言打下基础。

下面先通过几个简单的 C 语言程序例子,使大家对 C 语言程序有个初步认识。同时也列出 C++ 语言程序,使大家理解采用 C++ 语言进行 C 语言编程的意义,避免烦琐而意义不大的输入输出格式要求,更多关注于算法和程序实现。

### 例 1.3 输出一行文字。

```
#include<stdio.h>           //预处理命令,包含该头文件到此处
int main(void)              //主函数名,程序入口
{                            //函数开始标记
    printf("Hello World!\n"); //语句:输出字符
    return 0;
}                            //函数结束标记
```

上面是一个完整的 C 语言源程序,可以将其保存为 example1.c,运行后是在屏幕上输出一行文字,如图 1-4 所示。

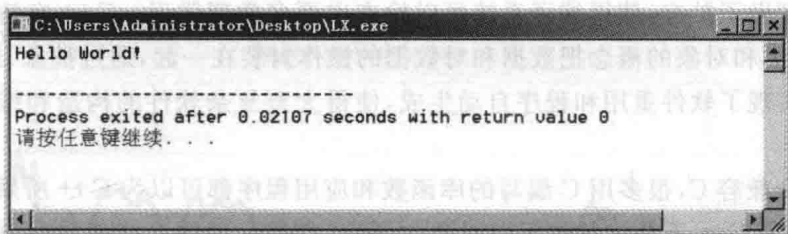


图 1-4 例 1.3 程序的运行结果

程序的第 1 行表示该程序要使用 stdio 库里的函数,所以将该库函数的头文件加入到程序前面。第 2 行是程序开始执行的入口,称为主函数(main),其返回类型为整型。

程序的主体就是一条语句,即输出字符,语句的结束处有分号“;”。printf 是 C 语言的格式输出函数,但其涉及变量时使用比较复杂。//表示 C 语言行注释,它及该行其后的内容编译时忽略。

注意,C 语言的程序语句中的符号如分号、引号、逗号、括号都是英文半角符号。

该程序改用 C++ 的输出流进行编写如下:

```
#include<iostream>
using namespace std;           //C++的命名空间
int main(void)
{
    cout<<"Hello World!";      //C++的输出流对象进行输出字符
    return 0;
}
```

该程序可以保存为 example1.cpp,程序运行结果相同。头文件有不同,程序多了表示命名空间的第 2 行,输出采用 cout 输出流对象的方法。

C 语言的程序包括头部和函数两部分。头部主要是一些编译预处理命令,而 C 语言程序的主体就是各个函数,必须有且只能有一个主函数,它是程序开始执行的入口。

**例 1.4** 输入圆的半径,输出该圆的周长和面积。

```
#include<stdio.h>
#define PI 3.14159              //宏定义
int main(void)
{
    double r,c,area;            //定义了 3 个变量
    printf("input radius : ");  //提示输入
    scanf("%lf",&r);           //输入半径
    c=2*PI*r;
    area=PI*r*r;                //运算
    printf("circumference is %f, area is %f.",c,area); //输出
    return 0;
}
```

运行后,如果输入半径 1,则输出圆周长和面积如图 1-5 所示。

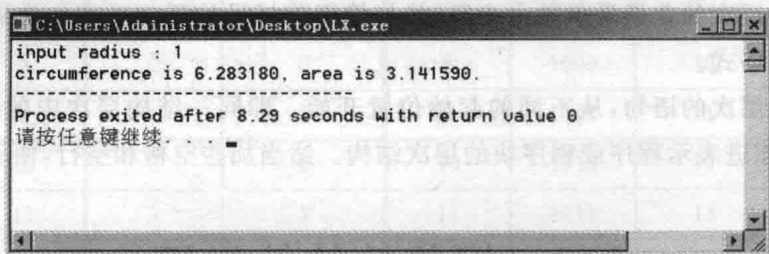


图 1-5 例 1.4 程序运行输入 1 后的结果

这里的头部多了一个宏定义,将 PI 定义为 3.14159,这样编译时将会把程序中的 PI 全部替换为 3.14159。从而避免反复输入圆周率常数,也便于在宏定义处进行修改,如高精度圆周率使用 10 位数,不必在程序中每处修改。

该程序改用 C++ 的输出流进行编写如下:

```
#include<iostream>
#define PI 3.14159
using namespace std;
int main(void)
{
    double r,c,area;
    cout<<"input radius : "; //提示输入
    cin>>r; //输入半径
    c=2 * PI * r;
    area=PI * r * r; //运算
    cout<<"circumference is "<<c<<" , area is "<<area<<". "; //输出
    return 0;
}
```

可见 C++ 的输入输出流对象进行输入输出要简单得多,没有类型差异,变量位置直观。

在 C/C++ 语言源程序中,凡是放在“/\* ... \*/”之间或以“//”开始到行尾的部分都是注释的内容,前者称为“块注释”,后者称为“行注释”。

程序在编译时,注释部分被忽略,不产生目标代码,对程序运行不起任何作用。但注释是程序的重要组成部分!注释可以提高程序的可读性,它是对编程意图、程序功能、算法、语句的作用等所做的必要说明,注释应在编程的过程中进行。对那些一目了然的内容一般在注释中就不必陈述,以免影响注释的效果。

C/C++ 语言程序在书写时不要将程序的每一行都由第一列开始,应在语句前面加进一些空格,称为“缩进”。

C/C++ 语言程序的书写格式自由度高,灵活性强,随意性大。如一行内可写一条语句,也可写几条语句;一条语句也可分写在多行内。因此,应该通过必要的缩进格式,或是在适当的地方加进一些空行,以提高程序的可读性,便于人们阅读和理解。