

从基础
到实践

基础知识全面覆盖
实践操作循序渐进

从理论
到应用

理论讲解详尽具体
动手应用实操实练

从入门
到进阶

内容编排由浅入深
进阶案例综合拓展

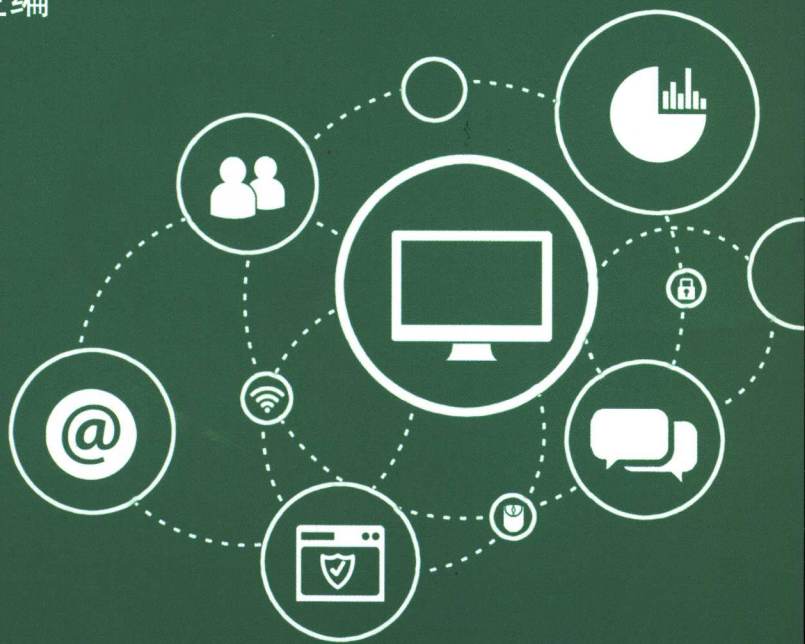
重点
推荐

C语言程序设计

案例教程



■ 付兴宏 李中跃◎主编



非
外
借

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

C 语言程序设计案例教程

主 编 付兴宏 李中跃
副主编 罗雨滋 孙 婷 李笑岩
许 悦 杨中兴
参 编 李寅杰 贾冬妮

内 容 简 介

C 语言是一门通用的基础性语言，应用非常广泛。本书根据 C 语言的知识脉络，采用典型的案例式教学模式展开知识陈述。本书文字浅显、案例详实，注重编程能力的培养，以一个实用的 C 语言项目为主线，循序渐进地讲解 C 语言程序的基本构成、C 语言的基本命令和功能、C 语言程序设计思想，以及利用 C 语言编写较复杂程序的基本方法。本书涵盖的知识主要包括 C 语言程序的基本构成、Visual C++ 2010 编程环境介绍、常用程序结构、数组和指针、结构体、文件以及综合项目开发等方面的内容。

本书重点突出编程思想和编程方法的介绍，内容精炼，表述清晰，实例丰富，以案例驱动的方式引导读者学习，操作性强，符合学习 C 语言的读者的需求。本书案例突出实用性，可以使读者通过实践具备举一反三的拓展能力。

版权专有 侵权必究

图书在版编目(CIP)数据

C 语言程序设计案例教程 / 付兴宏, 李中跃主编. --
北京: 北京理工大学出版社, 2022. 4

ISBN 978 - 7 - 5763 - 1053 - 5

I. ①C… II. ①付… ②李… III. ①C 语言 - 程序设计 - 教材 IV. ①TP312. 8

中国版本图书馆 CIP 数据核字(2022)第 030684 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68944723 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 三河市天利华印刷装订有限公司

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 20.25

字 数 / 450 千字

版 次 / 2022 年 4 月第 1 版 2022 年 4 月第 1 次印刷

定 价 / 94.00 元

责任编辑 / 钟 博

文案编辑 / 钟 博

责任校对 / 周瑞红

责任印制 / 施胜娟

图书出现印装质量问题，请拨打售后服务热线，本社负责调换

前言

C 语言具有灵活、高效、可移植等优点，是软件开发中最常用的计算机语言，它既有高级语言的优点，又有低级语言的诸多优点，既可以用来编写系统软件，又可以用来编写应用软件，目前广泛地应用在工业控制、智能仪表、嵌入式系统、硬件驱动、系统底层开发、中间件等领域。通过 C 语言的学习，可以很好地理解程序设计思想和学会程序设计的方法，因此 C 语言又成为学习其他高级语言的基础。学好 C 语言，对高职院校的学生来说是非常有益的。

本书以 Visual C++ 2010 学习版为编程环境，采用案例驱动方式，借鉴“学做一体”的理念编写而成。本书突出技能性和实用性，以培养编程能力为核心，以程序设计思想和编程方法为基础，重点介绍了程序设计基本结构、C 语言基本程序要素、实用程序开发等方面的知识。本书大部分内容以一个实际的应用程序项目为主线，涉及 C 语言程序构成和程序开发环境、C 语言基础、语言程序基本结构、函数、指针、结构体等几个部分。第 1 章主要介绍 C 语言程序的基本构成、C 语言开发环境和 C 语言综合项目的基本内容；第 2 章通过综合项目中数据处理功能的实现，主要介绍 C 语言中标识符、表达式等基本语言要素，数据的输入和输出等方面的知识；第 3 章通过综合项目菜单功能的实现，主要介绍 C 语言程序的基本结构及编程方法；第 4 章通过数据的排序和查找功能的实现，介绍 C 语言中数组的使用和编程方法；第 5 章通过综合项目的模块化实现，介绍 C 语言函数的编写方法；第 6 章主要介绍指针的概念和编程方法；第 7 章通过综合项目数据完整性表示功能的实现，主要介绍结构体的概念和使用方法；第 8 章通过综合项目数据存储功能的实现，主要介绍文件的概念和编程方法；第 9 章介绍了利用 C 语言设计和开发一个应用项目的步骤和过程，是对本书所介绍 C 语言知识和程序设计思想的总结，对明确 C 语言的应用起到了画龙点睛的作用。本书较全面地体现了 C 语言基本知识及编程思想的实际应用，涉及程序设计和计算机语言应用的大部分环节，结构清晰，应用实例丰富，实现了理论学习和具体应用的充分结合。

本书的特色如下。

- (1) 内容丰富，重点突出，覆盖当前 C 语言程序开发的基本要点，而且章节安排合理。
- (2) 应用案例驱动模式，每个章节围绕实际项目展开，强调教学内容的实用性。

(3) 强调实践教学, 实训内容明确可行, 便于高职院校学生理解和实践。

(4) 符合高职院校的教学特点, 结构由浅入深, 内容循序渐进。

(5) 给出了一个综合应用项目, 将全书的内容综合起来, 体现了 C 语言知识在一个综合应用项目设计的实现中的灵活应用。

(6) 项目训练和拓展训练自成系统, 通过项目训练的启发和提示, 学生能够更容易地实现拓展训练的内容, 从而使自己的 C 语言编程技能得到实质的提高。

(7) 以程序设计思想养成以及软件开发能力培养为中心, 为培养适应生产、服务第一线的技能型、应用型人才服务。

本书由付兴宏、李中跃担任主编, 付兴宏负责组织、筹划工作, 并亲自执笔。许悦、杨中兴、罗雨滋、孙婷、李笑岩担任副主编。付兴宏负责第 1、2、9 章的编写工作, 李中跃负责第 7 章的编写工作, 许悦负责第 5 章的编写工作, 杨中兴负责第 6 章的编写工作, 罗雨滋负责第 3 章和第 2 章部分节次的编写工作, 孙婷负责第 4 章的编写工作, 李笑岩负责第 8 章的编写工作。

由于编者水平有限, 书中难免有错误和疏漏之处, 敬请同行专家及广大读者批评指正。如读者在使用本书的过程中有其他意见或建议, 恳请向编者 (253024764@qq.com) 踊跃提出宝贵意见。

编者



在线课程资源

目录

第1章 C语言简介	1
1.1 项目概述	2
1.2 程序设计语言	2
1.3 C程序的基本构成	3
1.3.1 简单的C程序实例	3
1.3.2 C程序的构成和书写规则	5
1.4 C语言的字符集和关键字	6
1.4.1 C语言的字符集	6
1.4.2 C语言的词汇	6
1.5 C程序的运行环境	8
1.5.1 C程序的实现过程	8
1.5.2 熟悉 Visual C++2010 学习版编程环境	9
1.6 先导案例的设计与实现	15
1.6.1 问题分析	15
1.6.2 设计思路	16
1.6.3 程序实现	16
1.7 综合应用案例	17
1.7.1 Visual C++2010 学习版编程环境的使用	17
1.7.2 C程序的调试	17
习题	20
第2章 C语言数据类型及数据处理	21
2.1 C语言的基本数据类型	22
2.1.1 C语言的数据类型	22
2.1.2 基本数据类型及类型说明符	23
2.2 常量	25

2.2.1	整型常量	25
2.2.2	浮点型常量	26
2.2.3	字符型常量	27
2.2.4	字符串型常量	29
2.2.5	符号常量	29
2.3	变量	31
2.3.1	变量的定义和初始化	31
2.3.2	使用变量的注意事项	32
2.4	常用运算符及表达式	33
2.4.1	C 语言运算符和表达式概述	33
2.4.2	算术运算符	34
2.4.3	关系运算符	36
2.4.4	逻辑运算符	36
2.4.5	赋值运算符	37
2.4.6	自加、自减运算符	40
2.4.7	条件运算符	41
2.4.8	位运算符	42
2.4.9	逗号运算符	45
2.5	表达式中的类型转换	46
2.5.1	自动类型转换	46
2.5.2	强制类型转换	48
2.6	数据的输入和输出	49
2.6.1	C 语言语句分类	49
2.6.2	输入/输出概述	50
2.6.3	字符的输入/输出	51
2.6.4	格式输出函数——printf() 函数	52
2.6.5	格式输入函数——scanf() 函数	55
2.7	先导案例的设计与实现	59
2.7.1	问题分析	59
2.7.2	设计思路	59
2.7.3	程序实现	59
2.8	综合应用案例	60
2.8.1	交换两个变量值的问题	60
2.8.2	最大值和最小值问题	61
2.8.3	求一元二次方程根的问题	62
	习题	64

第3章 C 程序结构	65
3.1 三种基本程序结构	66
3.2 赋值语句	67
3.3 选择结构程序设计	68
3.3.1 if 语句	68
3.3.2 if 语句的嵌套	74
3.3.3 switch 语句	76
3.4 循环结构程序设计	79
3.4.1 循环结构的作用	79
3.4.2 几种循环语句及其比较	80
3.4.3 循环的嵌套	87
3.5 改变程序流程的几种语句	88
3.5.1 goto 语句	89
3.5.2 break 语句和 continue 语句	90
3.6 先导案例的设计与实现	92
3.6.1 问题分析	92
3.6.2 设计思路	92
3.6.3 程序实现	93
3.7 综合应用案例	95
3.7.1 三角形判定问题	95
3.7.2 成绩分级问题	96
3.7.3 最大公约数和最小公倍数问题	97
本章小结	99
习题	99
第4章 数组和字符串	101
4.1 一维数组	102
4.1.1 一维数组的定义和初始化	102
4.1.2 一维数组元素的引用	106
4.2 二维数组	107
4.2.1 二维数组的定义和初始化	107
4.2.2 二维数组元素的引用	110
4.3 字符数组和字符串	111
4.3.1 字符数组的定义和初始化	111
4.3.2 字符数组元素的引用	112
4.3.3 字符数组的输入和输出	112
4.3.4 常用字符串处理函数	115
4.4 先导案例的设计与实现	120

4.4.1 问题分析	120
4.4.2 设计思路	120
4.4.3 程序实现	120
4.5 排序的两种方法	121
4.5.1 冒泡排序法	122
4.5.2 选择排序法	124
4.6 综合应用案例	125
4.6.1 求 M 位学生 N 门课程的平均成绩问题	125
4.6.2 按字母顺序排列输出问题	127
本章小结	128
习题	129
第 5 章 函数和编译预处理	130
5.1 C 语言中的函数	132
5.1.1 模块化的含义	132
5.1.2 函数的分类	133
5.1.3 函数的定义	134
5.1.4 库函数	136
5.2 函数的参数和返回值	136
5.2.1 形参和实参	136
5.2.2 实参与形参之间的数据传递	137
5.2.3 数组作为函数参数的处理	138
5.2.4 函数的返回值	143
5.3 调用函数	144
5.3.1 函数调用形式	144
5.3.2 函数的嵌套调用	148
5.3.3 函数的递归调用	150
5.4 变量的作用域与存储类型	152
5.4.1 变量的作用域	152
5.4.2 变量的存储类型	158
5.5 内部函数与外部函数	164
5.5.1 内部函数	164
5.5.2 外部函数	164
5.6 编译预处理	166
5.6.1 宏定义	166
5.6.2 文件包含	170
5.6.3 条件编译	171
5.7 先导案例的设计与实现	173

5.7.1 问题分析	173
5.7.2 设计思路	174
5.7.3 程序实现	174
5.8 综合应用案例	178
本章小结	185
习题	186
第6章 指针	187
6.1 地址和指针	188
6.2 指针和指针变量	189
6.2.1 指针变量的定义	189
6.2.2 指针变量的赋值及初始化	190
6.2.3 直接访问和间接访问	191
6.2.4 取地址运算符和指针运算符	192
6.2.5 指针变量作为函数的参数	195
6.3 指针与数组	196
6.3.1 指向数组元素的指针变量	196
6.3.2 指向数组元素的指针变量的初始化	197
6.3.3 通过指针引用数组元素	197
6.3.4 指向数组的指针作为函数参数	207
6.4 指针与字符串	208
6.5 指针数组	211
6.6 指向函数的指针	214
6.6.1 用指向函数的指针变量调用函数	214
6.6.2 用指向函数的指针变量作函数参数	216
6.7 先导案例的设计与实现	217
6.7.1 问题分析	217
6.7.2 设计思路	218
6.7.3 程序实现	219
6.8 综合应用案例	220
6.8.1 遍历数组元素问题	220
6.8.2 字符串的连接问题	221
本章小结	222
习题	223
第7章 结构体与共用体	224
7.1 结构体	225
7.1.1 结构体的引出	225
7.1.2 结构体类型的定义	225

7.1.3	结构体类型变量的定义	226
7.1.4	关键字 typedef 的用法	227
7.1.5	结构体变量的引用和初始化	228
7.2	结构体数组	230
7.2.1	结构体数组的定义和初始化	230
7.2.2	结构体数组的应用实例	231
7.2.3	指针与结构体数组	232
7.3	结构体与函数	236
7.3.1	结构体变量的成员作参数	236
7.3.2	结构体变量作参数	237
7.3.3	指向结构体变量的指针作实参	237
7.4	共用体	239
7.4.1	共用体的定义	239
7.4.2	共用体变量的引用	240
7.5	先导案例的设计与实现	241
7.5.1	问题分析	241
7.5.2	设计思路	241
7.5.3	程序实现	242
7.6	综合应用案例	245
7.6.1	候选人得票统计问题	245
7.6.2	师生信息统计问题	247
	本章小结	249
	习题	251
第 8 章	文件系统	252
8.1	文件概述	253
8.1.1	理解文件的概念	253
8.1.2	文件的分类	254
8.1.3	文件的存取方式	254
8.2	文件的打开与关闭	254
8.2.1	文件的打开	254
8.2.2	文件的关闭	256
8.3	文件的读/写操作	257
8.3.1	文件的字符读/写操作	257
8.3.2	文件的块读/写操作	259
8.3.3	文件的字符串读/写操作	262
8.3.4	其他文件读/写函数	264
8.3.5	随机文件的读/写	266

8.3.6 出错检测	269
8.4 先导案例的设计与实现	270
8.4.1 问题分析	270
8.4.2 设计思路	271
8.4.3 程序实现	272
8.5 综合应用案例	277
8.5.1 统计文件内容问题	277
8.5.2 删除文件中的记录问题	279
本章小结	282
习题	282
第9章 C语言项目实例	283
9.1 系统分析	283
9.1.1 系统功能分析	283
9.1.2 系统数据分析	284
9.2 系统设计与实现	285
9.2.1 系统登录模块	285
9.2.2 系统初始化模块	286
9.2.3 系统录入模块	287
9.2.4 数据查询模块	291
9.2.5 数据修改模块	294
9.2.6 数据删除模块	298
9.2.7 数据打印模块	302
9.2.8 数据统计模块	303
9.2.9 其他数据模块	307
9.2.10 关于C语言知识的综合应用	311

第 1 章

C语言简介

—系统概述和环境安装—

【内容简介】

计算机中的软件系统是由程序语言开发的，一个应用系统的开发，离不开编程软件的支持。C语言是目前国际上最优秀的程序设计语言之一，它将程序代码与数据进行有效的分离，使程序结构更加清晰，同时也使程序的可读性以及可维护性变得更高。

本章首先描述学生管理系统项目的基本业务情况，其次介绍程序设计语言的基本内容，接着介绍C语言的结构特点、程序组成、书写规则、上机运行过程和调试应用程序的方法，最后描述C语言程序开发和运行环境的安装方法和步骤。

【知识目标】

了解C语言的发展和基本特点；理解C语言程序的构成、C语言的词法规定和书写规范；掌握C语言程序开发和运行环境的安装和使用方法。

【能力目标】

具备C语言程序的编写和调试能力；培养良好动手实践习惯；培养逻辑思维能力；培养学生发现问题和解决问题的自主学习能力。

【素质目标】

培养严谨分析、规范操作的工作素质；培养自主学习和思考的基本素质；树立编程为国的家国意识。

【先导案例】

应用程序的主要功能之一是提供人机交互界面，让用户通过操作实现工作要求。菜单界面是应用程序人机交互的主要界面，用户通过选择选项实现相应功能。在学生管理系统中，菜单界面是用户首先面对的界面，在功能菜单界面上显示各项功能（如信息的增加、修改、删除和查询）的对应选项，用户通过选择选项来实现不同的功能。通过本章内容的学习，我们要设计和实现学生管理系统的菜单界面，如图1-1所示。

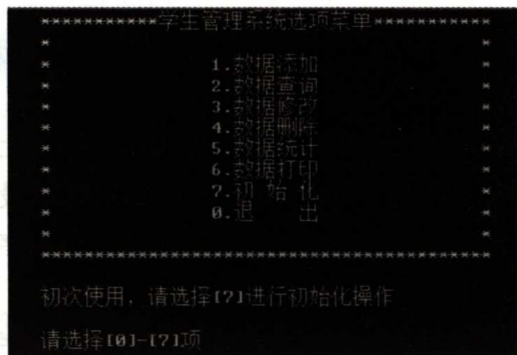


图 1-1 学生管理系统的菜单界面

1.1 项目概述

对一个学校来说,学生管理是一所学校重要的日常工作,学生信息管理是其中一项基本内容。开发一个学生管理系统,实现学生信息的计算机化管理,不仅方便、快捷,可以避免手工管理的诸多弊端,还可以将从事学生管理工作的教师从繁重的数据管理任务中解脱出来,使他们把更多的精力投入其他工作。

一个学校的学生信息主要包括学生的基本信息和学生的成绩信息两个项目,所以学生信息管理主要围绕这两个方面进行。

一般情况下,在学生信息管理工作伊始,学生管理人员首先要收集学生的基本信息,如学生的学号、姓名等,在考试之后,就可以依据学生的基本信息收集学生的成绩信息。一个简单的学生信息表见表 1-1。

表 1-1 一个简单的学生信息表

学号	姓名	班级	C 语言	网络原理	外语	总分	平均分
1	张蒙恬	网络一班	67	76	85	228	76
2	李思思	网络一班	65	87	88	240	80
3	王国庆	网络二班	88	98	92	278	92.7
4	欧阳兰	网络二班	79	90	95	264	88
5	赵晓丽	网络三班	68	86	78	232	77.3
6	佟瑞鑫	网络三班	66	77	88	231	77

在进行成绩处理之前,学生管理人员可以追加、修改和删除学生的基本信息,也可以查询学生的基本信息;在考试之后,学生管理人员可以添加、修改和删除学生的成绩信息,同时还要实现学生成绩排序和各分数段的统计工作。

1.2 程序设计语言

目前使用的计算机应用系统,如网上购物系统、办公系统、排版系统等,都是由计算机程序设计语言编写而成的。计算机程序设计语言通常简称为编程语言,是一组用来定义计算机程序的语法规则。程序设计语言让程序员能够准确地定义计算机所需要使用的数据,并精确地定义在不同情况下所应采取的行动。程序设计语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是程序设计语言。在程序设计语言中,这些记号串就是程序。程序设计语言有 3 个方面的因素,即语法、语义和语用。语法表示程序的结构或形式,亦即表示构成程序设计语言的各个记号的组合规律。语义表示程序的含义,亦即表示按照各种方法所表示的各个记号的特定含义。语用表示程序与使用者的关系。

最初的程序设计语言是一种用二进制代码“0”和“1”表示的、能被计算机直接识别和执行的语言，称为机器语言。它是一种低级语言，用机器语言编写的程序不便于记忆、阅读和书写。通常不用机器语言直接编写程序。

在机器语言的基础上，人们设计出了汇编语言，它将机器语言用便于人们记忆和阅读的助记符表示，如 ADD、SUB、MOV 等。计算机运行汇编语言程序时，首先将用助记符写成的源程序转换成机器语言程序才能运行。汇编语言适用于编写直接控制机器操作的低层程序，它与机器密切相关，汇编语言和机器语言都是面向机器的程序设计语言，称为低级语言。

随着计算机应用的发展，出现了高级程序设计语言。它是一种与硬件结构及指令系统无关，表达方式比较接近自然和数学表达式的计算机程序设计语言。

C 语言是一种灵活、高效、可移植性强的高级程序设计语言。1972—1973 年，贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言，后来又对 C 语言进行多次改进。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的使用，成为当代最优秀的程序设计语言。

C 语言是一种结构化语言。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强，它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构，还可以直接访问内存的物理地址，进行位（bit）一级的操作。由于 C 语言实现了对硬件的编程操作，因此 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，也可用于应用软件的开发。此外，C 语言还具有效率高、可移植性强等特点。

用高级语言编写的程序称为“源程序”。源程序不能被计算机识别和执行，需要将其翻译成机器指令，翻译方式通常有编译和解释两种。

编译方式是将源程序整个编译成等价的、独立的目标程序，然后通过链接程序将目标程序链接成可执行程序。解释方式是将源程序逐句翻译，翻译一句执行一句，边翻译边执行，不产生目标程序，在整个执行过程中，解释程序都一直在内存中。

1.3 C 程序的基本构成

1.3.1 简单的 C 程序实例

用 C 语言编写的程序称为 C 程序或 C 源程序。下面先介绍两个简单的 C 程序，从中分析 C 程序的特性。这几个程序都是在 Visual C++ 2010 Express 环境下编译通过的。

【应用案例 1.1】

用 C 语言编写一个程序，输出“你好，欢迎使用 C 语言!”。

【程序代码】

```
/* ex1_1.c: 输出欢迎词 */  
#include <stdio.h>
```

```

void main( )                /* 定义主函数 */
{
    printf("你好,欢迎使用 C 语言! \n");    /* 输出欢迎词 */
}

```

【程序运行结果】

程序运行结果如图 1-2 所示。



图 1-2 应用案例 1.1

程序运行结果

【程序说明】

(1) 程序中的 main() 代表一个函数, 其中 main 是函数名,

void 表示该函数的返回值类型。main() 是一个 C 程序中的主函数, 程序从主函数开始执行。一个 C 程序有一个且只能有一个主函数 main()。一个 C 程序可以包含多个文件, 每个文件又可以包含多个函数。函数之间是相互平行、相互独立的。执行程序时, 系统先从主函数开始运行, 其他函数只能被主函数调用或被主函数所调用的函数调用。

(2) 函数体用 “{}” 括起来。main() 函数中的所有操作语句都在这 “{}” 之中。

(3) “#include <stdio.h>” 是一条编译预处理命令, 声明该程序要使用 “stdio.h” 文件中的内容, “stdio.h” 文件包含输入函数 scanf() 和输出函数 printf() 的定义。编译时系统将头文件 “stdio.h” 中的内容嵌入程序中该命令位置。C 语言的编译预处理命令都以 “#” 开头。C 语言提供了 3 类编译预处理命令: 宏定义命令、文件包含命令和条件编译命令。应用案例 1.1 中出现的 “#include <stdio.h>” 是文件包含命令, 其中尖括号内是被包含的文件名。

(4) 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, printf() 函数的功能是把要输出的内容送到显示器显示, 双引号中的内容要原样输出。“\n” 是换行符, 即在输出完 “你好, 欢迎使用 C 语言!” 后回车换行。

(5) 每条语句以 “;” 号结束。

(6) “/*……*/” 部分是一段注释, 注释只是为了改善程序的可读性, 是对程序所需部分的说明, 向用户提示或解释程序的意义。“/*” 是注释的开始符号, “*/” 是注释的结束符号, 必须成对使用。程序编译时, 不对注释作任何处理。注释可出现在程序中的任何位置。

【应用案例 1.2】

输入两个正整数, 计算并输出两数的和。

【程序代码】

```

/* ex1_2.c:求两个正整数的和 */
#include <stdio.h>
void main( )                /* 主函数 */
{
    int a,b,sum;            /* 定义 3 个整型变量 */
    printf("请输入两个正整数! \n");
}

```

```

scanf("%d",&a);          /* 输入数据给变量 a */
scanf("%d",&b);          /* 输入数据给变量 b */
sum = a + b;             /* 变量 a 和变量 b 的值相加,然后将结果赋给变量 sum */
printf("相加结果是%d\n",sum); /* 输出变量 sum 的值 */
}

```

【程序运行结果】

程序运行结果如图 1-3 所示。

【程序说明】

(1) “int a, b, sum;”是变量声明。变量是程序运行过程中存储数据的内存单元,此处声明了3个整数类型的变量 a, b, sum。C语言的变量必须先声明后使用。

(2) 程序中的 scanf 是输入函数的名字,程序中的 scanf()函数的作用是输入 a, b 的值。&a 和 &b 中的“&”的含义是取地址,此 scanf()函数的作用是将两个数值分别输入变量 a 和 b 的地址所标志的单元,也就是输入给变量 a 和 b。

(3) “sum = a + b;”是将 a, b 两变量的内容相加,然后将结果赋值给整型变量 sum。

(4) “printf (“相加结果是%d \n”, sum);”是调用库函数 printf()输出 sum 的结果。“%d”为格式控制符,表示 sum 的值以十进制整数的形式输出。

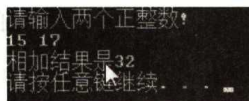


图 1-3 应用案例 1.2
程序运行结果

1.3.2 C程序的构成和书写规则

1. C程序的构成

(1) C程序是由函数构成的,函数是C程序的基本单位。一个源程序至少包含一个 main()函数,即主函数,也可以包含一个 main()函数和若干个其他函数。被调用的函数可以是系统提供的库函数,也可以是用户根据需要自己设计编写的函数。

(2) main()函数是每个程序执行的起始点,一个C程序不管有多少个文件,有且只能有一个 main()函数。一个C程序总是从 main()函数开始执行,而与 main()函数在程序中的位置无关。可以将 main()函数放在整个程序的最前面,也可以放在整个程序的最后,或者放在其他函数之间。

(3) 源程序可以有预处理命令(include 是其中一种),预处理命令通常放在源文件或源程序的最前面。

(4) 每个语句都必须以分号结尾,但预处理命令、函数头和花括号“}”之后不加分号。

(5) 标识符和关键字之间至少加一个空格以示间隔,空格的数目不限。

(6) 源程序中需要解释和说明的部分,可用“/*……*/”加以注释,注释是给程序阅读者看的,机器在编译和执行程序时,注释将被忽略。

2. C程序的书写规则

从书写清晰,便于阅读、理解、维护的角度出发,养成良好的编程风格,在书写程序时应遵循以下规则。

(1) 在C语言中,虽然一行可写多个语句,一个语句也可占多行,但是为了便于阅读,