

普通高等教育“十三五”规划教材

C 语言程序设计 项目化教程

主编 袁园 钱新杰

副主编 胡桂香 刘春晓 李彪



国防工业出版社

National Defense Industry Press

III

C 语言程序设计 项目化教程

主 编 袁 园 钱新杰
副主编 胡桂香 刘春晓 李 彪

国防工业出版社

·北京·

内 容 简 介

本教材主要使用 VC6.0 集成开发工作作为教学平台。教材内容以标准 C 语言为基础,突出 C 语言的基础性和实践中的应用性,注重程序设计编码规范和方法。内容由浅入深,层次结构清晰,案例化教程。教材采用模块化教学模式,主要由 C 语言程序设计基础、基本数据类型和表达式、程序流程的控制、数组、函数、指针、结构体与共用体和文件八个模块组成。所有知识都结合具体实例进行介绍,涉及的程序代码给出了详细的注释,可以使读者轻松领会使用 C 语言进行程序开发的精髓与乐趣,快速提高实际开发技能。

本教材非常适合作为编程初学者的学习用书,也适合作为开发人员的查阅、参考资料。

图书在版编目(CIP)数据

C 语言程序设计项目化教程 / 袁园, 钱新杰主编. —
北京: 国防工业出版社, 2016. 8
ISBN 978 - 7 - 118 - 11020 - 3

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

中国版本图书馆 CIP 数据核(2016)第 203058 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

三河市鼎鑫印务有限公司印刷

新华书店经售

*

开本 787 × 1092 1/16 印张 16 字数 368 千字

2016 年 8 月第 1 版第 1 次印刷 印数 1—3000 册 定价 38.50 元

(本书如有印装错误,我社负责调换)

国防书店:(010)88540777

发行邮购:(010)88540776

发行传真:(010)88540755

发行业务:(010)88540717

C 语言程序设计是高等职业教育计算机专业的专业基础课程,也是 IT 技术从业人员的必备技能之一。C 语言具有概念简洁、数据类型丰富、表达能力强、使用灵活方便、目标程序高、可移植性好、控制流和数据结构新颖、程序结构性和可读性好等优点,有利于培养读者良好的编程习惯。C 语言是介于汇编语言和高级语言之间的一种通用的、模块化编程语言,程序代码效率高,既具有高级语言的特点,又具有低级语言的特点,既适合于编写系统软件,又适合于编写应用软件,应用面非常广。现代一些优秀的编程语言,如 Java、C++、C#、Perl、PHP、JavaScript 等都深受 C 语言的影响,有了 C 语言基础后,再学习和使用其他编程语言将变得相当轻松。

高等职业教育注重提高内涵发展,教学质量的水平是内涵发展的重要内容,本书依据高等职业教育以“以学生为中心、以能力为本位”的基本教育理念,重视学生实践能力的基本原则,在内容组织上更趋合理,教材内容展现形式更加简单、准确。本教材从初学者的角度出发,以通俗易懂的语言,丰富多彩的实例,详细介绍了使用 C 语言进行程序设计应该掌握的各方面知识。

本教材的编写人员都是拥有多年丰富 C 语言教学经验的任课老师,多年的高职学生教学,培养了很多职业化应用的学生。本书是通过不同知识点的讲解,以及学生不同的反馈,总结修订的结果。

目 录

模块一 程序设计基础	1
项目一 概述	1
任务一 程序和程序设计	1
任务二 程序设计语言	2
任务三 程序设计方法	3
项目二 C 程序设计的初步认识	4
任务一 C 语言简介	4
任务二 标识符	5
任务三 注释符	6
任务四 C 语言程序的书写规则	6
任务五 C 语言程序的基本结构	6
项目三 程序设计过程	8
任务一 解决问题的基本步骤	8
任务二 C 语言程序的设计过程	8
项目四 算法	9
任务一 算法的概念	9
任务二 算法的描述方式	10
任务三 简单算法举例	11
项目五 开发环境简介	12
任务一 启动 Visual C++ 6.0	12
任务二 创建工程	12
任务三 创建文件	14
任务四 编辑程序	15
任务五 编译与连接	16
任务六 运行	16
任务七 关闭	16
习题	16
模块二 基本数据类型和表达式	17
项目一 计算机中数据的编码与运算	17

	任务一 计算机内存	17
	任务二 编码的概念	18
项目二	数据类型	18
项目三	常量和变量	19
	任务一 常量	19
	任务二 变量	24
项目四	运算符与表达式	29
	任务一 算术运算符和算术表达式	30
	任务二 赋值运算符和赋值表达式	31
	任务三 自增与自减运算符	35
	任务四 关系运算符和关系表达式	38
	任务五 逻辑运算符和逻辑表达式	39
	任务六 逗号表达式和条件表达式	42
	任务七 位运算符	44
	任务八 其他运算符	47
项目五	常用的输入/输出库函数	48
	任务一 printf() 函数	48
	任务二 scanf() 函数	52
	任务三 其他输入输出函数	53
项目六	数学函数	54
项目七	随机数发生器函数	54
	习题	56
模块三 程序流程的控制		58
项目一	程序的基本结构	58
项目二	C 语句分类	59
项目三	顺序结构	61
项目四	选择结构	62
	任务一 C 语言的逻辑值	62
	任务二 if 语句	63
	任务三 switch 语句	72
	任务四 程序应用举例	75
项目五	循环结构	79
	任务一 while 语句	79
	任务二 do-while 语句	82
	任务三 for 语句	84
	任务四 循环结构的嵌套	86
项目六	转向语句	91
	任务一 break 语句	91

任务二	continue 语句	92
项目七	程序范例	94
	习题	96
模块四	数组	100
项目一	数组概述	100
项目二	一维数组	100
任务一	一维数组的定义	100
任务二	一维数组初始化	102
任务三	一维数组元素的引用	102
任务四	引用中要注意的问题	105
任务五	一维数组应用举例	108
项目三	二维数组	117
任务一	二维数组的定义	117
任务二	二维数组初始化	118
任务三	二维数组元素的引用	119
任务四	二维数组应用举例	120
项目四	字符数组	123
任务一	字符串与字符数组	123
任务二	字符数组的定义	123
任务三	字符数组的初始化	124
任务四	字符数组的引用	124
任务五	字符串的输入和输出	125
任务六	字符串处理函数	129
任务七	字符数组应用举例	131
项目五	字符串数组	133
	习题	136
模块五	函数	140
项目一	库函数	140
项目二	函数的定义和声明	142
任务一	函数的定义	142
任务二	函数的返回值	143
任务三	函数的声明	145
项目三	函数的调用	146
任务一	函数调用一般形式	146
任务二	函数调用方式	146
任务三	函数调用过程	147
项目四	函数的参数	148

项目五	函数的嵌套调用和递归调用	151
	任务一 函数的嵌套调用	151
	任务二 函数的递归调用	151
项目六	编译预处理命令	155
	任务一 文件包含	155
	任务二 宏定义	155
	任务三 条件编译	158
	习题	161
模块六	指针	164
项目一	指针的概念	164
	任务一 地址与指针	164
	任务二 指针变量的定义	166
	任务三 指针变量的赋值	168
	任务四 指针变量的引用	170
	任务五 指针变量的基本运算	172
项目二	指针与函数	178
	任务一 指针作函数的参数	178
	任务二 函数的返回值为指针	180
	任务三 指向函数的指针	181
项目三	指针与数组	183
	任务一 指向数组的指针变量的定义	183
	任务二 一维数组与指针	184
	任务三 二维数组与指针	188
项目四	指针与字符串	194
	任务一 用字符指针变量处理字符数组	194
	任务二 用字符指针变量处理字符串常量	197
	任务三 用字符指针数组处理多个字符串	198
	习题	201
模块七	结构体与共用体	205
项目一	结构体的概念	205
	任务一 结构体类型的定义	205
	任务二 结构体变量的定义与初始化	206
	任务三 结构体变量的引用	209
项目二	结构体数组	213
	任务一 结构体数组的定义	214
	任务二 结构体数组的引用	215
	任务三 结构体数组的初始化	216

任务四 结构体数组的应用·····	216
项目三 共用体·····	218
任务一 共用体类型的定义和变量定义·····	219
任务二 共用体类型变量的引用·····	220
任务三 共用体类型举例·····	221
习题·····	221
模块八 文件 ·····	224
项目一 文件的基本概述·····	224
任务一 文件概述·····	224
任务二 文件指针·····	225
项目二 文件的打开与关闭·····	225
任务一 文件的打开·····	226
任务二 文件的关闭·····	227
项目三 文件的读写·····	228
任务一 字符输入输出函数·····	228
任务二 字符串输入输出函数·····	230
任务三 数据块输入输出函数·····	230
任务四 格式化输入输出函数·····	231
项目四 文件读写位置的定位·····	234
任务一 rewind() 函数·····	234
任务二 fseek() 函数·····	234
任务三 ftell() 函数·····	235
项目五 文件结束检测·····	235
习题·····	237
附录一 C 语言的关键字 ·····	239
附录二 常用字符与 ASCII 代码对照表 ·····	240
附录三 运算符的优先级和结合性 ·····	241
附录四 双目算术运算中两边运算数类型转换规律 ·····	243
附录五 库函数 ·····	244
参考文献 ·····	248

计算机所能完成的工作以及怎样完成工作都是由人们事先编好的程序来控制的。程序设计是人们为了通过计算机实现某一功能或解决问题,而使用某种程序设计语言编写程序并实现结果的过程。设计计算机程序常常借助于程序设计工具,如 Visual C++ 6.0 集成开发环境。

C 程序设计是众多计算机程序设计中的一种,学习使用 C 语言编写程序,可以帮助我们建立一种程序设计的思维方式,并掌握运用 C 程序设计来解决某些实际问题的处理过程。

通过本模块的学习,读者应掌握程序设计的基本概念,能用 C 语言设计一个简单的程序,熟悉 Visual C++ 6.0 开发工具的使用。

项目一 概述

任务一 程序和程序设计

自从 1946 年世界上第一台计算机诞生以来,计算机的应用范围已经渗透到社会的各个领域,计算机改变了世界,也改变了人类的生活方式。但是,计算机的强大功能和灵活的自动处理却是由人类编制的程序控制的。人们按照既定目标设计和编写程序,并输入到计算机中,计算机通过执行这些程序,才能逐一实现预期的目标和得到需要的结果。

计算机程序是指为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。计算机本身是不会做任何工作的,它是按照程序中的有序指令来完成相应任务的。只有用事先编好的程序去控制它,才能让它为人服务。人们为了完成某项具体的任务而编写一系列指令,并将这一系列指令交给计算机去执行,这个过程称为程序设计。

计算机能完成各种复杂的任务,不熟悉计算机的人会把它想象得十分神秘。其实,计算机不过是一种具有内部存储能力、由程序自动控制的电子设备。计算机执行每一个操作都是按照人们事先指定的内容和步骤进行的。人事先规定出计算机完成某项工作的操作步骤,每个步骤的具体内容由计算机能够理解的指令或语句来描述,这些指令或语句将告诉计算机“做什么”和“怎样做”。

在用计算机处理问题前要考虑很多的细节,写程序时把每个细节的执行过程都要描述清楚。所有问题在编写程序前都要考虑清楚,明确告诉计算机“怎样做”。计算机的功

能固然强大,但更重要的是程序设计人员开发出的各种简单或复杂的程序。正是这些程序,使计算机更好地为人们服务。

任务二 程序设计语言

已经知道程序是由指令序列组成的,告诉计算机如何完成一个具体的任务。由于现在的计算机还不能理解人类的自然语言,所以还不能用自然语言编写计算机程序。人们要使用计算机,让计算机按人们的意志进行工作,就必须让计算机能理解并执行人们给它的指令。这就需要找到一种人和计算机都能识别的语言——程序设计语言。程序设计语言是用计算机能够理解的语言来表达所设计程序的含义,是人与计算机之间进行交流和通信的工具。

人们可以使用各种程序设计语言编写计算机程序,其中某些程序指令能够直接被计算机执行,而另一些程序指令还需要通过中间的翻译处理以后,才能够被计算机执行。因此,我们将计算机程序语言分为机器语言、汇编语言和高级语言3类。对于计算机本身来说,它并不能直接识别由高级语言编写的程序,它只能接受和处理由0和1的代码构成的二进制指令或数据。

1. 机器语言

机器语言是以二进制代码的形式来表示基本的指令集合,每条指令均为0和1组成的二进制代码串。用机器语言编写的程序计算机能直接识别和执行,运算速度快,占用的存储空间小。但机器语言存在明显的局限性:

由于机器语言与计算机界硬件的设计密切相关,因此所有的计算机只能直接执行其自身的机器语言建立的程序。在研制某一型号的计算机时,要事先设计好该型号计算机的机器语言系统,规定好每条由若干个0和1组成的机器指令能使计算机实现何种操作,即特定的机器语言只能用在特定的一类计算机上。由于机器语言的指令系统是由二进制数码表示,所以用它编写程序既难学、难写,又难于理解和维护,而且不同型号的计算机的机器语言是不同的,因此用机器语言编写的程序不能通用。

2. 汇编语言

为了改善机器语言难记、难写等特点,汇编语言用指令英文名称的缩写作为助记符代替机器的操作指令,用标号和符号来表示地址、常量和变量。用汇编语言编写的程序与用机器语言编写程序的方式相似,它们的指令是一一对应的。因为计算机只能识别机器指令,汇编语言编写的程序不能被计算机直接识别和执行,需要通过转换过程处理成机器语言指令,为此人们又开发了称为“汇编程序”的“翻译程序”,实现将符号化的汇编语言转换成机器指令,才能被计算机执行。

汇编语言便于识别和记忆,执行效率也比较高,但用汇编语言编写的程序通用性差,而且工作量大,仍然不能让人满意。于是,出现了高级语言。

3. 高级语言

高级语言中的语句一般采用类似人类自然语言中的自然词汇,使得程序更容易阅读和理解。高级语言提供了丰富的数据类型和运算符,语句功能强大,一条语句往往相当于多条指令。高级语言还独立于具体的硬件系统,使得程序的通用性、可移植性和编写程序的效率大大提高。

针对各种应用领域,人们设计了很多种高级语言,其中 C 语言既具有其他高级语言的优点,又具有低级语言的许多特点。它功能丰富,移植性强,编译质量高,成为应用最广泛的高级语言之一。同样,计算机也不能直接识别用高级语言编写的程序,也需要把高级语言程序转换成机器语言指令,这种转换由编译器提供。不同的高级语言需要不同的编译程序,采用的翻译方法也可能不同,主要有两种方式:

1) 编译方式

首先针对具体高级语言(例如 C 语言)开发出一个编译软件,将该种语言程序翻译为所用计算机机器语言的等价程序,即目标程序(二进制代码的形式)。最后由计算机执行这个机器语言程序。

2) 解释方式

首先针对具体高级语言(例如 BASIC 语言)开发一个解释软件,该软件读入高级语言程序,并能一步步地按照程序要求工作,完成程序所描述的计算。这种解释软件边解释语句边执行,不生成目标程序。

任务三 程序设计方法

程序设计方法对程序设计的质量有非常重要的影响。程序设计方法先后经历了三个主要阶段:非结构化程序设计阶段、结构化程序设计阶段和面向对象的程序设计阶段。

非结构化程序设计方法由于其诸多的弊端,现已基本淘汰。目前,程序设计的方法有两大类:面向过程的结构化程序设计方法和面向对象的程序设计方法。

1. 结构化程序设计

结构化程序设计方法是基于模块化、自顶向下、逐步细化和结构化编码等程序设计技术而发展起来的。它强调程序设计风格和程序结构的规范化,提倡清晰的结构,有助于程序设计思想的形成和理解。C 语言就属于结构化程序设计语言。

面向过程的结构化程序设计的核心思想是功能的分解。解决实际问题时,首先要做的工作就是将问题分解为若干个功能模块。其基本思路是,对实际问题进行分析,先从整体出发,把一个较复杂的问题分解为若干个相对独立的子功能,对每个子功能可以再细化为若干个低一层的子功能,直到子功能便于在计算机上实现。这就是自顶向下、逐步细化的解决问题的方法。结构化程序设计方法主要使用顺序、选择和循环三种基本的控制结构来实现这些功能模块。

2. 面向对象程序设计

面向对象技术是当今比较流行的软件设计与开发技术,主要包括面向对象分析、面向对象设计、面向对象编程、面向对象测试以及面向对象软件维护等。面向对象程序设计技术的提出,主要是为了解决传统程序设计方法所不能解决的代码重用问题。

面向对象程序设计是建立在结构化程序设计基础之上的程序设计方法,其最重要的变化是程序设计围绕数据来设计,而不是围绕操作本身。在面向对象程序设计中,将程序设计为一组相互协作的对象,而不是一组相互协作的函数。

对象是面向对象技术中的一个重要概念。所谓“对象”就是对客观存在事物的一种表示,是包含现实世界物体特征的抽象实体,是物体属性和行为的一个组合体。从程序设计的角度看,对象是指将数据和使用这些数据的一组基本操作封装在一起的统一体,它是

程序的基本运行单位,具有一定的独立性,其他对象的操作不能操作该对象隐藏起来的数据。

面向对象程序设计中的另一个重要概念是类。类是具有相同操作(功能)和相同数据(属性)的集合。类是对一类具有相同特征和行为事物(对象)的抽象表示。对象是某个类的具体实现。这样,面向对象程序设计以类作为构造程序的基本单位,具有封装、数据抽象、继承、多态性等特征。

项目二 C 程序设计的初步认识

C 程序设计是众多计算机程序设计中的一种,学习使用 C 语言编写程序,可以帮助我们建立一种程序设计的思维方法,并掌握运用 C 程序设计来解决某些实际问题的处理过程。

任务一 C 语言简介

C 语言是贝尔实验室于 20 世纪 70 年代初研制出来的,后来又被多次改进,并出现了多种版本。当代最优秀的程序设计语言早期的 C 语言主要是用于 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,到了八十年代,C 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。C 语言是一种广泛应用的计算机程序设计语言,也是国际上公认的最重要的几种通用程序设计语言之一。它可用来编写系统软件,也可用来编写应用软件。C 语言作为 UNIX 操作系统的开发语言而广为人们认识,当今许多操作系统都是用 C 语言或 C++ 语言编写的,世界各地的程序员都使用它来编写各种程序。目前,C 语言已经逐渐地应用于绝大多数的计算机中。

C 语言在各种计算机上的快速推广导致了许多 C 语言版本的出现,这些版本虽然是类似的,但通常不完全兼容,这令程序员很头痛。为解决这个问题,美国国家标准化组织(ANSI)于 1983 年成立了一个委员会,以确定 C 语言的标准定义——ANSI 标准 C 语言。现代的 C 语言编译器绝大多数都遵守这一标准。

作为一种简短、清楚、高效的程序设计语言,C 语言是应用最广泛的语言之一。它具有以下主要特点:

1. 语言简洁紧凑,使用方便灵活

C 语言仅有 32 个关键字,9 种控制语句。C 程序表达方式简洁,书写形式自由,主要由小写字母表示。它把高级语言的基本结构和语句与低级语言的实用性结合起来。

2. 运算符丰富

C 语言包含 34 种运算符,范围广泛。它把括号、赋值、逗号等都作为运算符处理,使得运算类型极其丰富,表达式类型多样化,可以实现其他高级语言难以实现的运算。

3. 数据类型丰富

C 语言的数据类型有整型、实型、字符型、枚举型和各种构造类型,还允许用户自定义类型。利用这些数据类型可以实现复杂的数据结构,完成各种类型的数据处理。

4. C 语言是结构化程序设计语言

C 语言语法限制不太严格,程序设计自由度大,程序编写者有较大的自由度。它具有

结构化的控制语句,包含顺序、选择、循环三种基本结构,结构清晰。函数是构成 C 语言程序的基本单位,用函数作为程序模块实现程序的模块化,使程序便于使用和维护。

5. 效率高,可移植性好

与汇编语言相比,C 语言适用范围更广泛。它可用于多种操作系统,也适用于多种机型,可直接对硬件进行操作。C 语言允许直接访问物理地址,能像汇编语言一样对计算机最基本的位、字节和地址进行操作。它既有高级语言的优点,又有低级语言的许多功能;既可以编写系统软件,又可以编写各种应用软件。

基于上述特点,在当今众多的计算机编程语言中,C 语言被认为是最受欢迎的语言之一。学习好 C 语言可作为程序设计语言学习的良好开端。

任务二 标识符

标识符是用来标识程序中的变量、常量、数据类型、数组、函数等的名称,是合法的字符序列,合法标识符的命名必须满足以下语法规则:

- (1) 标识符只能由字母、数字和或下划线组成。
- (2) 标识符的第一个字符必须是字母或下划线。
- (3) 标识符区分大小写字母,例如 a 和 A 是两个不同的标识符。

合法的标识符如:

a3, student_1, _test

不合法的标识符如:

3a(开头是数字), f-6(有“-”), s 5(有空格), d6.4(有“.”)

对于标识符的长度,C 语言编译系统是有规定的,即标识符的前若干个字符有效,超过的字符将不被识别。不同的 C 语言编译系统所规定的标识符有效长度可能会不同。有些系统允许取较长的名字,读者在取名时应当了解所用系统的具体规定。

C 语言的标识符可以分为以下三类。

1. 关键字

关键字又称保留字,是预先定义的、具有特殊意义的标识符。用户不能重新定义与关键字相同的标识符,也不能把关键字作为一般标识符使用。因为,编译器会按照固定含义来解析这些关键字。C 语言中的关键字请参考附录一。

2. 预定义标识符

预定义标识符是指在 C 语言中预先定义并具有特定含义的标识符,如 C 语言提供的库函数的名字(如 printf)和预编译处理命令(如 define)等。C 语言允许把这类标识符重新定义另作他用,但这将使这些标识符失去预先定义的原意。由于目前各种计算机系统的 C 语言都把这类标识符作为固定的库函数名或预编译处理命令使用,为了避免误解,建议用户不要把这些预定义标识符另作他用。

3. 用户标识符

由用户根据需要定义的标识符称为用户标识符,又称为自定义标识符。用户标识符一般用来给变量、函数、数组等命名。程序中使用的用户标识符除要遵守标识符命名规则外,还应注意做到“见名知意”,一般选择有一定含义的英文单词或汉语拼音作为标识符,如 number、red、day 等,以增加程序的可读性。

用户标识符不能与 C 语言的关键字相同,最好也不能和预定义标识符相同。如果用户标识符与关键字相同,则在对程序进行编译时系统将给出出错信息;如果用户标识符与预定义标识符相同,系统并不报错,只是该预定义标识符将失去原定含义,取而代之的是用户确定的含义,这样有可能会引发一些运行时错误。

任务三 注释符

在编写程序时可以在程序中加入注释,以说明变量的含义、语句的作用和程序段的功能,从而帮助人们阅读和理解程序。因此,一个好的程序应该有详细的注释。C 程序注释有如下两种:

1. /* 注释内容 */

此格式用于注释一个程序块,注释内容可以是英文字母、数字及汉字的一行或多行。

2. //注释内容

此格式用于注释一行,在一行中“//”后面的内容都将被注释。

注释可出现在程序中的任何位置,它是为便于编程者和阅读程序的人理解程序的意义而设置的。注释仅是写给人看的,而不是写给编译器看的。程序编译时,不对注释作任何处理,目标代码中不含注释。在学习和编写程序过程中,需要养成良好的标注注释的习惯。另外在调试程序中对暂不使用的语句也可用注释符括起来,使编译器跳过不作处理,待调试结束后再去掉注释。

任务四 C 语言程序的书写规则

C 语言是自由格式语言,一行可以写多个语句,一条语句也可以写在多行,在书写时可随意安排格式,如换行、添加空格等,格式变化不影响程序意义。为了便于阅读和维护程序,更好地体现程序的层次结构,书写所采用的通用规则是:

(1) 程序一般用小写字母书写。

(2) 一行一般写一条语句。

(3) 在程序里适当加入空行,分隔程序的不同部分。

(4) 同层次不同部分对齐排列,下一层次的内容通过适当退格(在一行开始加空格),使程序结构更清晰。

(5) 在程序里增加一些说明性信息即添加注释。

开始学习程序设计时就应养成良好的书写习惯,以便能够写出结构清晰、便于阅读理解和维护的程序。

任务五 C 语言程序的基本结构

通过下面的例子可以说明 C 语言程序的结构。

【例 1-1】一个简单的 C 语言程序示例。

```
#include <stdio.h>          /* 预处理命令,注明使用的库函数 */
void main ()                /* 主函数从 main()开始 */
{
    printf("This a c program! \n"); /* 从标准输出设备输出字符串 */
}
```

该程序的运行结果如图 1-1 所示。



图 1-1 例 1-1 运行结果

从这个例子可以看到,一个 C 语言程序的基本结构分为两个部分:

一部分是以 # 号开头预处理部分(程序中第一行),说明程序中要用到 C 语言系统提供的库函数(如程序中用到 printf() 输出函数)时,要将所对应的头文件(如 stdio.h)包含进来。注意这里用的是尖括号“<>”。所有的 C 编译程序都提供能完成各种基本任务的函数的集合——函数库,程序员在编程过程中需要用到这些基本的函数时只需从函数库中直接调用即可。

另一部分称为函数(程序中其余几行),它是程序的基本部分,用于描述程序所完成的工作。程序中的 main() 函数称为主函数。在 C 语言中,规定程序总是从 main() 函数开始执行,因此,该函数相当于程序的入口。main() 函数中包含有用户自己编写的指令代码,也可以通过调用和控制其他函数进行协同工作,如上例中调用了 printf() 函数来实现在屏幕上打印字符的功能。需要强调的是,一个程序只能有一个 main() 函数。

一个 C 语言的函数由返回值类型(void 表示没有返回值)、函数名、参数列表和函数体组成,具体的结构将在后面章节详细描述。函数体的语法与 main() 函数的语法相同。

该程序的运行结果就是输出一行信息“This a c program!”。程序运行结果如图 1-1 所示(图中最后一行显示的“Press any key to continue”是 Visual C++ 6.0 自动添加的)。

如果在 main() 函数中需要调用其他的函数,而且这些函数不是系统提供的,就需要用户自己编写,这些函数可以接着 main() 函数书写,如例 1-2 所示。

【例 1-2】输入长方体的长、宽、高,计算长方体体积。

```
#include <stdio.h>
void main ()
{
    int x,y,z,v;                /* 定义长、宽、高、体积为整型变量 */
    printf("请输入长方体的三边长:"); /* 显示输入提示 */
    scanf("% d,% d,% d",&x,&y,&z); /* 调用标准函数,要求从键盘输入 x、y、z 的值 */
    v=x*y*z;                    /* 计算体积 v */
    printf("v=% d\n",v);        /* 在屏幕上输出结果 */
}
```

运行时首先要求用户输入 3 个整数,然后程序自动计算出其体积,如图 1-2 所示。

由此可以看出:C 程序有且只有一个主函数 main(),C 程序的函数可以包括由编译系统提供的标准函数(如 printf()、scanf() 等)和由用户自己定义的函数。C 语言程序的结构具有以下特点:

- (1) 一个 C 语言程序可以由一个或多个文件组成。
- (2) 程序中可以有预处理命令(如 include 命令),预处理命令通常放在程序的最


```
请输入长方体的三边长: 3,4,5
V=60
Press any key to continue
```

图 1-2 例 1-2 运行结果

前面。

(3) C 语言程序是由一个或多个独立的函数构成的,函数是构成 C 语言程序的基本单位。

(4) 一个 C 语言程序必须包含一个主函数 `main()`,也只能有一个主函数。程序从 `main()` 函数开始执行,也在 `main()` 结束。

(5) 函数包含两部分,一个是函数的首部;另一个是函数体。函数的首部包括函数名、函数类型和用圆括号“()”括起来的形参说明;函数体是用花括号“{ }”括起来的函数执行部分。

(6) 函数体中是各种语句,语句是程序的基本执行单位。每一个语句都以分号“;”作为结束。但预处理命令,函数头和花括号“}”后不能加分号。

项目三 程序设计过程

任务一 解决问题的基本步骤

一般来说,用计算机解决问题的过程包含以下几个部分。

(1) 分析问题。设计一种解决方案,通过程序语言严格描述这个解决方案。

(2) 确定数据结构。根据任务书提出的要求,指定的输入数据和输出结果,确定存放数据的数据结构。

(3) 确定算法。针对存放数据的数据结构来确定解决问题、完成任务的步骤。有关算法的概念将在下一节中介绍。

(4) 编码。根据确定的数据结构和算法,使用选定的计算机语言编写程序代码,输入到计算机并保存在磁盘上,简称编程。

(5) 调试程序。消除由于疏忽而引起的语法错误或逻辑错误;用各种可能的输入数据对程序进行测试,使之对各种合理的数据都能得到正确的结果,对不合理的数据能进行适当的处理。

(6) 整理并写出文档资料。

任务二 C 语言程序的设计过程

利用 C 语言开发一个程序,一般包括以下几个步骤。

(1) 分析问题,确定程序目标。明确程序需要做什么,并设计解决问题的步骤。仔细考虑程序需要的信息、要进行的计算和操作、要反馈的结果信息。