



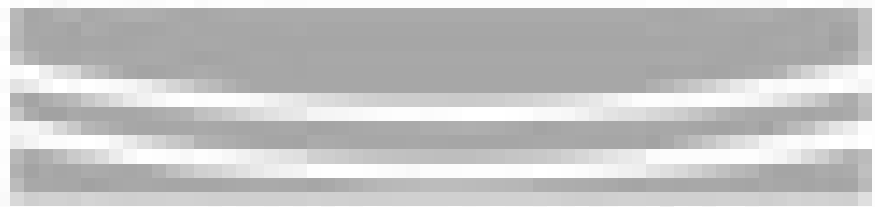
THE UNIVERSITY OF CHINA PRESS

# 中国思想史论

第一卷

先秦两汉思想史论

作者：[Name] 著



ISBN: [Number]

北京：[Publisher] 2000年



新世纪高职高专计算机基础教育系列规划教材

# C 语言 程 序 设 计

(第五版)

高职高专教材编审委员会组编

主 编 董汉丽 副主编 韩彦峰 蒋高飞 孙俊岭



C YUYAN CHENGXU SHEJI

大连理工大学出版社  
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

## 图书在版编目(CIP)数据

C 语言程序设计 / 董汉丽主编. —5 版. —大连:大连理工大学出版社, 2009. 1

新世纪高职高专计算机基础教育系列规划教材

ISBN 978-7-5611-2000-2

I. C… II. 董… III. C 语言—程序设计—高等学校: 技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 007421 号

大连理工大学出版社出版

地址:大连市软件园路 80 号 邮政编码:116023

发行:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail:dutp@dutp.cn URL:http://www.dutp.cn

大连理工印刷有限公司印刷 大连理工大学出版社发行

---

幅面尺寸:185mm×260mm 印张:18 字数:410 千字

印数:67901~72900

2002 年 3 月第 1 版

2009 年 1 月第 5 版

2009 年 1 月第 10 次印刷

---

责任编辑:马 双

责任校对:白 俊

封面设计:张 莹

---

ISBN 978-7-5611-2000-2

定 价:29.80 元

# 总 序

我们已经进入了一个新的充满机遇与挑战的时代,我们已经跨入了 21 世纪的门槛。

20 世纪与 21 世纪之交的中国,高等教育体制正经历着一场缓慢而深刻的革命,我们正在对传统的普通高等教育的培养目标与社会发展的现实需要不相适应的现状作历史性的反思与变革的尝试。

20 世纪最后的几年里,高等职业教育的迅速崛起,是影响高等教育体制变革的一件大事。在短短的几年时间里,普通中专教育、普通高专教育全面转轨,以高等职业教育为主导的各种形式的培养应用型人才的教育发展到与普通高等教育等量齐观的地步,其来势之迅猛,发人深思。

无论是正在缓慢变革着的普通高等教育,还是迅速推进着的培养应用型人才的高职教育,都向我们提出了一个同样的严肃问题:中国的高等教育为谁服务,是为教育发展自身,还是为包括教育在内的大千社会?答案肯定而且惟一,那就是教育也置身其中的现实社会。

由此又引发出高等教育的目的问题。既然教育必须服务于社会,它就必须按照不同领域的社会需要来完成自己的教育过程。换言之,教育资源必须按照社会划分的各个专业(行业)领域(岗位群)的需要实施配置,这就是我们长期以来明乎其理而疏于力行的学以致用问题,这就是我们长期以来未能给予足够关注的教育目的问题。

如所周知,整个社会由其发展所需要的不同部门构成,包括公共管理部门如国家机构、基础建设部门如教育研究机构和各种实业部门如工业部门、商业部门,等等。每一个部门又可作更为具体的划分,直至同它所需要的各种专门人才相对应。教育如果不能按照实际需要完成各种专门人才培养的目标,就不能很好地完成社会分工所赋予它的使命,而教育作为社会分工的一种独立存在就应受到质疑(在市场经济条件下尤其如此)。可以断言,按照社会的各种不同需要培养各种直接有用人才,是教育体制变革的终极目的。

随着教育体制变革的进一步深入,高等院校的设置是否会同社会对人才类型的不同需要一一对应,我们姑且不论。但高等教育走应用型人才培养的道路和走研究型(也是一种特殊应用)人才培养的道路,学生们根据自己的偏好各取所需,始终是一个理性运行的社会状态下高等教育正常发展的途径。

高等职业教育的崛起,既是高等教育体制变革的结果,也是高等教育体制变革的一个阶段性表征。它的进一步发展,必将极大地推进中国教育体制变革的进程。作为一种应用型人才培养的教育,它从专科层次起步,进而应用本科教育、应用硕士教育、应用博士教育……当应用型人才培养的渠道贯通之时,也许就是我们迎接中国教育体制变革的成功之日。从这一意义上说,高等职业教育的崛起,正是在为必然会取得最后成功的教育体制变革奠基。

高等职业教育还刚刚开始自己发展道路的探索过程,它要全面达到应用型人才培养的正常理性发展状态,直至可以和现存的(同时也正处在变革分化过程中的)研究型人才培养的教育并驾齐驱,还需要假以时日;还需要政府教育主管部门的大力推进,需要人才需求市场的进一步完善发育,尤其需要高职教学单位及其直接相关部门肯于做长期的坚忍不拔的努力。新世纪高职高专教材编审委员会就是由全国100余所高职高专院校和出版单位组成的旨在以推动高职高专教材建设来推进高等职业教育这一变革过程的联盟共同体。

在宏观层面上,这个联盟始终会以推动高职高专教材的特色建设为己任,始终会从高职高专教学单位实际教学需要出发,以其对高职教育发展的前瞻性的总体把握,以其纵览全国高职高专教材市场需求的广阔视野,以其创新的理念与创新的运作模式,通过不断深化的教材建设过程,总结高职高专教学成果,探索高职高专教材建设规律。

在微观层面上,我们将充分依托众多高职高专院校联盟的互补优势和丰裕的人才资源优势,从每一个专业领域、每一种教材入手,突破传统的片面追求理论体系严整性的意识限制,努力凸现高职教育职业能力培养的本质特征,在不断构建特色教材建设体系的过程中,逐步形成自己的品牌优势。

新世纪高职高专教材编审委员会在推进高职高专教材建设事业的过程中,始终得到了各级教育主管部门以及各相关院校相关部门的热忱支持和积极参与,对此我们谨致深深谢意,也希望一切关注、参与高职教育发展的同道朋友,在共同推动高职教育发展、进而推动高等教育体制变革的进程中,和我们携手并肩,共同担负起这一具有开拓性挑战意义的历史重任。

新世纪高职高专教材编审委员会

2001年8月18日

# 前 言

《C 语言程序设计》(第五版)是新世纪高职高专教材编委会组编的计算机类课程规划教材之一。

由于 C 语言的重要性,当前高职院校都把 C 语言程序设计作为入门程序设计课程来对待。我们针对高职高专学生自身的特点与培养目标,从高职高专学生的思维方式、理解能力以及在后续课程中的应用等诸方面的因素出发,对《C 语言程序设计》(第四版)教材进行了改革和修订。我们的目标是修订一本具有新体系、新结构的通俗易懂、结构新颖的 C 语言实用教材。

本教材的任务是,训练学生掌握计算机程序设计的基本操作技能;使学生具有程序设计的能力,并掌握用计算机解决和处理问题的思维方法,为进一步学习和应用计算机其他专业知识打下良好基础。同时为进行嵌入式编程的开发与应用打下基础。

参加编写本教材的教师都具有长期讲授 C 语言的教学经验,这次修订内容体现了他们在教学中对 C 语言研究的一些成果以及丰富的教学实践经验;同时充分吸取了不少高职院校的意见和建议,在保持原有写作风格的基础上,有了如下新特色:

(1)分散难点,循序渐进。基础知识和理论体系删繁就简,容易入门和理解。对难以理解和容易出错的地方,还附加有醒目的说明与提示。为了使学生能在有限的时间内学会并掌握本门课程的知识,提高编程和实践能力,我们将指针一章的内容合理化解,分散到相关章节中去。这样,降低了学习指针的难度,使学生由浅入深,轻松地掌握、理解指针的用法。解决与摆脱了以往教材中将指针独立成章,教师不好教、学生不好学的难题。

(2)结构合理,通俗易懂。本教材结构上紧紧把握逻辑上的严密性,章节合理、脉络清晰、环环相扣;语言上简明扼要、文字流畅、通俗易懂。



#### 4 / C 语言程序设计 □

(3)精选例题与习题,注重理论与实际结合。修订了原教材中的部分例题、习题和技能训练。本教材的每一章都配有大量的习题、综合练习题及上机实训的内容。为了便于学生自学,提高编程的实践技能,我们对所有习题都通过调试,并进行了详尽的分析、提示及解答。本教材有配套的《C 语言程序设计习题解答与技能训练》指导教材及电子教案。

(4)增加了图形、动画的内容。因为图形的直观性可以激发、调动学生的学习及编程兴趣,进而可提高学生的动手实践能力,本教材中的部分图形动画出于学生之手。

本教材由董汉丽任主编,韩彦锋、蒋高飞、孙俊玲任副主编。具体编写分工如下:董汉丽编写第 1 章;董洁编写第 2 章;孙俊玲编写第 3、11 章;牟淑杰编写第 4、5、7 章;蒋高飞编写第 6、9 章;韩彦锋编写第 8、10 章;齐云飞编写附录。全书由董汉丽统稿。

尽管我们在《C 语言程序设计》(第五版)教材的特色建设方面做了大量的探索与尝试,但是,不足之处仍在所难免。我们衷心希望使用本教材的各教学单位和读者能够提出积极的改进意见,使我们在下次修订时更加完善。

为方便教师教学和学生自学,本教材配有答案和课件,如有需要,请登录我们的网站下载。

所有意见和建议请发往:gzjckfb@163.com

欢迎访问我们的网站:<http://www.dutpgz.cn>

联系电话:0411-84707492 0411-84706104

编 者

2009 年 1 月

# 目 录

---

<b>第 1 章 C 语言概述</b> .....	1
1.1 C 语言的发展与特点 .....	1
1.2 C 语言的基本符号与关键字 .....	3
1.3 C 语言程序 .....	5
1.4 Turbo C 集成环境与程序的上机操作 .....	7
习 题 .....	12
<b>第 2 章 数据类型、表达式及其运算</b> .....	14
2.1 C 语言的数据类型 .....	14
2.2 常 量 .....	16
2.3 变 量 .....	19
2.4 运算符及表达式 .....	23
2.5 赋值运算符和赋值表达式 .....	30
2.6 指针运算 .....	31
2.7 其他运算符及其表达式 .....	37
习 题 .....	40
<b>第 3 章 顺序结构程序设计</b> .....	43
3.1 结构化程序设计的基本概念及 C 语句 .....	43
3.3 数据的输入和输出 .....	45
3.4 程序设计举例 .....	53
习 题 .....	55
<b>第 4 章 选择结构程序设计</b> .....	59
4.1 关系运算符和关系表达式 .....	59
4.2 逻辑运算符和逻辑表达式 .....	60
4.3 条件语句 .....	61
4.4 多分支语句 .....	66
4.5 程序设计举例 .....	68
习 题 .....	71



<b>第 5 章 循环结构程序设计</b> .....	74
5.1 while 语句 .....	74
5.2 do-while 语句 .....	75
5.3 for 语句 .....	77
5.4 三种循环的比较 .....	78
5.5 多重循环 .....	79
5.6 循环中用到的其他语句 .....	79
5.7 程序设计举例 .....	81
习 题 .....	86
<b>第 6 章 数组和指向数组的指针</b> .....	89
6.1 数组的概念 .....	89
6.2 一维数组 .....	90
6.3 二维数组 .....	93
6.4 字符型数组与字符串 .....	96
6.5 指针与数组 .....	105
6.6 指针数组 .....	111
6.7 指向指针的指针变量 .....	113
6.8 程序设计举例 .....	114
习 题 .....	118
<b>第 7 章 函数与指针</b> .....	122
7.1 概 述 .....	122
7.2 函数的定义 .....	124
7.3 函数的参数和函数的值 .....	125
7.4 函数的调用 .....	127
7.5 函数的嵌套调用 .....	130
7.6 函数的递归调用 .....	131
7.7 指针与函数 .....	133
7.8 变量的作用域与存储类别 .....	148
7.9 内部函数和外部函数 .....	155
7.10 宏定义 .....	157
7.11 程序举例 .....	159
习 题 .....	167
<b>第 8 章 结构体与共用体</b> .....	170
8.1 结构体类型的定义 .....	170
8.2 定义结构体类型的变量 .....	171

8.3 结构体类型变量的引用 .....	173
8.4 结构体变量的初始化 .....	175
8.5 结构体数组 .....	176
8.6 指向结构体类型数据的指针 .....	178
8.7 用指针处理链表 .....	182
8.8 共用体 .....	192
8.9 枚举类型 .....	194
8.10 类型定义 .....	195
8.11 程序举例 .....	196
习 题 .....	199
<b>第 9 章 文 件</b> .....	<b>202</b>
9.1 文件概述 .....	202
9.2 文件的打开与关闭 .....	204
9.3 文件的读写 .....	206
9.4 文件定位与随机读写 .....	211
9.5 文件的出错检测 .....	213
9.6 程序举例 .....	213
习 题 .....	217
<b>第 10 章 C 语言屏幕操作函数及应用</b> .....	<b>218</b>
10.1 字符屏幕操作函数 .....	218
10.2 Turbo C 的图形与动画处理 .....	221
10.3 程序举例 .....	240
习 题 .....	248
<b>第 11 章 C 语言综合实训</b> .....	<b>252</b>
<b>附 录</b> .....	<b>266</b>
附录 I 常用字符与 ASCII 代码对照表 .....	266
附录 II Turbo C 库函数 .....	267
<b>参考文献</b> .....	<b>276</b>

# 第 1 章

## C 语言概述

### 本章要点

- C 语言的发展与特点
- C 语言的基本符号和关键字
- C 语言程序结构及特点
- C 语言程序编辑、编译与运行

## 1.1 C 语言的发展与特点

### 1.1.1 C 语言的发展

#### 1. 概述

C 语言是目前世界上最为流行的计算机高级程序设计语言之一。它设计精巧,功能齐全,既适合于编写应用软件,又特别适合于编写系统软件。据统计,PC 机领域许多著名的系统软件和应用软件都是运用 C 语言加上汇编语言子程序编写而成的。

C 语言不仅有易学易用的优点,而且具有面向过程、结构化、通用性强、可移植性强及集成开发环境先进等特点。而且,C 语言具有很强的绘图能力、数据处理能力,同样适合于二维、三维图形和动画的开发。

掌握程序设计的前提是掌握程序设计语言。在众多的程序设计语言中,C 语言以其灵活性和实用性受到了广大计算机应用人员的喜爱。几乎任何一种机型、任何一种操作系统都支持 C 语言程序开发。C 语言在其原有应用领域的基础上,又拓展了新的应用领域,支持大型数据库开发和 Internet 应用以及嵌入式系统的应用。

嵌入式系统代表着目前最新科技水平,也是当前最热门、最有发展前途的 IT 业应用领域之一。目前,嵌入式系统的应用已经从家用电器扩展到汽车、医疗、航天以及军事系统等领域中。因此,市场对具有“嵌入式”技能的编程者的需求大增,许多软件开发正在转向这个领域。

要想应对嵌入式系统技能特有的挑战,就需要学习 C 语言程序设计。所有嵌入式系统中需要的关键的方法技巧,都详细地包含在 C 程序设计内,包括端口管脚的控制以及开关的读取等。应用最为广泛的是嵌入式 C 语言编程、Linux C 语言编程。有些嵌入式

## 2 / C语言程序设计 □

应用系统平台是用 C、C++ 进行搭建、驱动(Linux)及应用(Java)的。

C 语言有很多分支,如:TC、BC、MSC 等,其结构大同小异。本书主要讲解 Turbo C。

### 2. C 语言的发展

早期的计算机,都是用机器语言和汇编语言来编写程序代码。到了第二代计算机才有了高级语言。

1960 年开发的 ALGOL-60,对其后的高级语言的发展起到了很好的推进作用。但是,它是一种面向问题的语言,过于抽象,难以描述系统。因此,没有得到真正的推广。

1963 年,英国剑桥大学推出了 CPL(Combined Programming Language)语言。它比 ALGOL-60 更接近于硬件,但其规模较大,难以实现和学习。

1967 年,英国剑桥大学的 Martin Richards 对 CPL 语言进行了简化,推出了 BCPL(Basic Combined Programming Language)语言。

1972 年~1973 年间,贝尔实验室的 D. M. Ritchie 在 BCPL 语言的基础上设计出了 C 语言。

1973 年,K. Thompson 和 D. M. Ritchie 两人合作,把原来由两人用汇编语言编写的 UNIX 操作系统中 90% 以上的代码用 C 语言来重写,即 UNIX 5。后来,C 语言虽做了多次改进,但主要还是用在贝尔实验室内部。直到 1975 年,用 C 语言编写的 UNIX 6 公布后,才引起业内人士的广泛关注。

1978 年,Brian W. Kernighan 和 Dennis M. Ritchie(合称 K&R)合著了影响深远的著作《The C Programming Language》一书。该书中介绍的 C 语言被称为标准 C。

以后,相继出现了很多 C 语言版本,如 Microsoft C、Turbo C、Quick C、Borland C 等。它们在语法上基本是相同的,在函数数量和功能上有较大的区别,使用时要注意区分。

1983 年,美国国际标准化协会(ANSI)在参考 C 语言各种版本的基础上,制定了新的标准,称为 ANSI C。

1988 年,K&R 按照 ANSI C 标准重写了《The C Programming Language》。1990 年,国际标准化组织 ISO(International Standard Organization)接受了以 87 ANSI C 为 ISO C 的标准。目前流行的 C 编译系统都是以它为基础的。

高级语言发展至今,面向对象的程序设计方法越来越受到人们的青睐。比如:Visual FoxPro(VFP)、Visual Basic(VB)、Visual C++(VC++)、C++、Java、C# 等等。其中,功能比较强大的还是 C++ 语言,它以 C 语言为基础,在很多方面二者兼容。因此,掌握了 C 语言,再进一步学习 C++ 或其他面向对象的程序设计语言如 Java、C# 等就相对容易多了。

本书以目前流行的 Turbo C 2.0 为开发环境,以 ANSI C 为基础,全面介绍 C 语言及其程序设计。

### 1.1.2 C 语言的特点

C 语言之所以能够在与众多的高级语言竞争中脱颖而出,成为高级语言中的佼佼者,这主要是因为与普通高级语言相比,它具有以下特点:

(1)语言简洁、灵活。C 语言只有 32 个标准的关键字、45 个标准的运算符以及 9 种

控制语句。程序书写形式自由,主要运用小写字母表示。

(2)数据类型丰富,涉及面广。C 语言的数据类型包括:整型、实型、字符型、枚举类型;结构体、共用体、数组和文件类型;指针类型;空类型。其中,整型、实型、字符型中还有多种小的类型。指针类型是 C 语言中最具特点的一种数据类型,它使用起来非常灵活,把 C 语言的功能特点发挥得淋漓尽致。

(3)运算符多样,表达能力强。C 语言中有 44 个运算符。除包括算术、关系、逻辑等常规运算符之外,还含有指针、地址、位、自增自减、条件、复合赋值运算符,甚至连圆括号、方括号、逗号、小数点等都能够用运算符来表达。由于 C 语言的运算符类型极为丰富,所以能够实现各种各样的高级和低级运算。

(4)函数是程序的主体。C 语言中,函数是程序的基本单位。用函数作构件,可以设计开发出结构清晰、功能齐全的大型程序。

(5)C 语言允许直接访问物理地址。C 语言中的位运算和指针运算能够直接对内存地址进行访问操作,可以实现汇编语言的大部分功能,即直接对硬件进行操作。

(6)生成的目标代码质量高。C 语言简洁、紧凑,程序执行速度快。它比一般的高级语言生成的目标代码质量高约 20%,只比汇编语言低 10%~20%。在高级语言中是出类拔萃的。

(7)可移植性好。C 语言所提供的与硬件有关的操作,如数据输入、输出等,都是通过调用系统提供的库函数来实现的。库函数本身并不是 C 语言的组成部分。因此,用 C 语言编写的程序能够很容易地从一种计算机环境移植到另一种计算机环境中。

正是由于 C 语言具有其他语言不可比拟的这些优点,使得它越来越受到程序设计人员的重视,并在众多的领域里得到广泛的应用。

任何事物都是一分为二的,C 语言也有其不足之处:其一,运算符的优先级和结合性比较复杂,不容易记忆。其二,C 语言的语法限制不太严格。比如,数组下标不作超界检查;整型、字符型、逻辑型可以通用,这些程序设计的灵活性在一定程度上降低了安全性。所以,这就对程序设计人员提出了更高的要求。

## 1.2 C 语言的基本符号与关键字

### 1.2.1 基本符号集

字符集是构成 C 语言的基本元素。用 C 语言编写程序时,所写的语句是由字符集中的字符构成。C 语言的字符集由下列字符构成:

(1)英文字母:A~Z,a~z。

(2)数字字符:0~9。

(3)特殊符号:

空格 ! % \* & \\_ (下划线) + = - ~

< > / \ ' " ; . , ( ) [ ] { }

### 1.2.2 关键字

关键字是具有特定含义的、专门用来说明 C 语言的特定成分的一类单词、单词缩写或单词组合。C 语言的关键字都是用小写字母来书写的。例如,用关键字来定义整型变量 a,应该写为 int a,而不能写成 INT a。由于关键字具有特定的含义和用途,所以,不能随便用于其他场合。否则,就会产生编译虽然能通过,但是运行结果却不对的现象,而且不容易检查出错误。以下列出 C 语言的关键字:

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	unsigned	union	void	volatile	while

### 1.2.3 标识符

C 语言中标识符(名字)是用来标识变量名、符号常量名、函数名、数组名、类型名等程序对象的有效字符序列。C 语言对标识符有如下规定:

(1)标识符只能由英文字母(A~Z, a~z)、数字(0~9)和下划线三种字符组成,且第一个字符必须为字母或下划线。

例如:正确的标识符:abc, a1, prog\_to, file\_2, ab2c3

非法的标识符:2from, yes no/

(2)大小写字符代表不同的标识符。例如:标识符 abc 与标识符 ABC 是两个不同的标识符。一般对变量命名用小写,对符号常量命名用大写。

(3)每个标识符由 1~8 个字符组成。如果标识符超过 8 个字符长度,只有前 8 个字符为有效标识符。例如:employee1 和 employee2 就会被认为是相同的标识符 employee。

**注意:**ANSI C 没有限制标识符长度。但是,各个编译系统都有自己的规定和限制, Turbo C 2.0 限制 8 个字符,超出的部分将被系统忽略。

(4)不能使用 C 语言的关键字作为标识符。建议标识符命名应当有一定的意义,做到“见名知义”。例如:length(表示长度)、sum(表示和)、time(表示时间)、pi(表示圆周率)。

### 1.2.4 ASCII 字符集

计算机中,所有的信息都用二进制代码表示。二进制编码的方式较多,应用最为广泛的是 ASCII 码。我们使用的字符在计算机中就是以 ASCII 码方式存储的。

ASCII 码是美国标准信息交换码(American Standard Code for Information Interchange,缩写为 ASCII)。它被国际标准化组织(ISO)认定为国际标准,详见附录 I。

ASCII 码分为标准 ASCII 码和扩展 ASCII 码。标准 ASCII 码在内存中占用一个字节,字节中的低 7 位用于编码,因此,可以表示 128 个符号。基本字符 0~9、A~Z、a~z 等编码值为 32~127,控制符的编码值为 0~31。控制符用于计算机向外部设备输出一些特殊的命令,如控制打印机换行、换页等。

## 1.3 C语言程序

### 1.3.1 程序与C程序结构

#### 1. 程序

所谓“程序”，即从事某一件事情的具体步骤。现实生活中，我们每个人其实都在无意识的情况下编写着不同的“程序”。例如：我们去银行取钱，就会有以下的操作步骤：

第一步，拿卡或存折去某一个银行；

第二步，在银行排队等候；

第三步，排队轮到自己后，把卡或存折交由银行工作人员进行取钱操作；

第四步，处理完后，银行工作人员将卡或存折以及现金返还给我们；

第五步，将卡或存折以及现金收放好后离开银行。

上面所描述的五个步骤就是我们去银行取钱的程序。所谓操作步骤，也就是执行某一件事情的先后顺序。

计算机的“程序”是人们编写的计算机代码的指令集合。计算机本身不会做任何操作，它所有的操作都是按照人们设计的计算机“程序”的顺序来执行的。不同的计算机语言有不同的语法规则和语句结构。下面，我们讲述C语言的程序结构。

#### 2. C程序结构

通过以下几个例题，让我们对C语言的程序结构及执行过程有个全面的了解。程序语句不要求完全能理解，先对C程序的全貌有一个感性认识，为以后的学习打下基础。

**【例 1.1】** 在计算机屏幕上输出信息。

```
/* 程序名称: simple. c */
#include <stdio. h>                /* 包含文件 */
main()                             /* 程序从函数 main 开始执行 */
{
    printf("I am a C programmer, I like C!"); /* 在屏幕上输出引号内的信息 */
}                                   /* 函数 main 结束 */
```

上述程序是一个简单而完整的C语言程序。经过编辑、编译和连接，程序运行后显示字符串：I am a C programmer, I like C!

说明：

(1) /\* … \*/ 之间的内容是语句的注释部分，供阅读程序用。计算机并不执行注释部分的内容。

(2) 语句 #include <stdio. h> 是包含文件。C语言中数据的输入、输出都是通过调用系统提供的库函数 scanf 和 printf 等来实现的。这些函数的说明都包括在 stdio. h 文件中。所以一般C程序中，文件头都需要有 #include <stdio. h>、#include <math. h> 等语句。

(3) main 是函数的名称，又称主函数。用 {} 括起来的内容是函数体，函数体由若干条语句组成。这是计算机要执行的部分，每条语句要以分号“;”结束。

## 6 / C 语言程序设计 □

(4)printf 是打印函数,作用是在屏幕上输出引号内的信息。

### 【例 1.2】 求两数之和。

```
#include <stdio. h>          /* 包含文件 */
main()                       /* 定义主函数 */
{
    int a,b,sum;             /* 定义局部变量 a,b,sum */
    scanf("%d,%d",&a,&b);    /* 输入,2 个整数,存入变量 a 和 b 中 */
    sum=a+b;                 /* 将 a,b 的和送入 sum */
    printf("sum=%d\n",sum);  /* %d 表示数据按十进制整型数输出结果 */
}
```

程序运行:输入 10,20 ↵ (回车键)

结果:sum=30

此程序从 main 函数开始执行,执行到 scanf 语句时,等待用户从键盘输入两个整型数据后,再继续执行。假如用户输入 10 和 20(此时,a 的值为 10,b 的值为 20),最后,由 printf 语句打印出结果:sum=30。

### 【例 1.3】 求两个数的最大值。

```
int max(int x,int y)         /* 定义函数 max,求两个数的最大值 */
{
    int z;                   /* 定义变量 z */
    if(x>y)z=x;              /* 进行条件判断,将最大值赋给变量 z */
    else z=y;
    return(z);               /* 返回函数值 */
}
#include <stdio. h>          /* 包含文件 */
main()                       /* 主函数 */
{
    int a,b,c;               /* 定义变量 a,b,c 为整型 */
    scanf("%d,%d",&a,&b);    /* 输入 a,b 的值 */
    c=max(a,b);              /* 调用 max 函数,将函数返回的最大值赋给变量 c */
    printf("max=%d\n",c);    /* 打印 c 的值 */
}
```

程序运行:输入 10,20 ↵

结果显示:max=20

程序由主函数 main 和求两个数的最大值的自定义函数 max 组成。此程序从 main 函数开始执行,执行到 scanf 语句时,等待用户从键盘输入两个整型数据后,再继续执行。假设我们输入 10 和 20,按回车键后,系统自动赋给 a 和 b,开始调用 max 函数。执行 max 函数时,把 a 和 b 的值传递给形参 x,y,然后求两个数的最大值。计算结束,退出 max 函数,返回到 main 函数,将返回的最大值送给变量 c。最后,由 printf 语句打印出结果:max=20。



### 1.3.2 C语言程序结构的特点

通过上面的几个例题,我们可以看出,C语言的程序结构具有如下特点:

(1)一个C程序可以有多个函数,但只能有一个名为main的主函数。程序从main()开始,通过函数之间的调用来实现相应的功能,最后以main()结束。

(2)主函数或子函数的函数体必须用一对花括号“{}”括起来。函数体中的语句将实现程序的预定功能。

(3)每条语句的最后必须有一个分号“;”。

(4)大小写表示不同的含义,程序语句一般用小写字母书写,大写一般用作符号常量。

(5)注释语句“/\* \*/”是非执行语句,可以放在程序的任意位置。在程序中加上注释可以提高程序可读性。编程时,为了使程序易读,应养成加注释的习惯。

(6)C语言中使用的所有变量都必须先定义为某种数据类型,然后才能使用。

(7)#include命令是编译预处理命令。其作用是将由双引号或尖括号括起来的文件中的内容读入到该语句位置处。在使用C语言输入、输出库函数时,一般需要使用#include命令将stdio.h文件包含到源文件中。一个#include命令只能指定一个包含文件。如果要包含多个文件,就需要用多个#include命令。

## 1.4 Turbo C集成环境与程序的上机操作

### 1.4.1 Turbo C集成编程环境

Turbo C 2.0是Borland公司开发的微机上的一个C语言集成开发环境。可以在Turbo C中完成C语言程序的编辑、编译、连接、运行、调试。

- 编辑 用高级语言或汇编语言编写的程序称为源程序。将C语言源程序通过键盘输入到计算机中,并以文件的形式存入到磁盘的过程称为“编辑”。经过编辑后得到的源程序文件是以“.c”为扩展名的文件。

- 编译 源程序经过“编译程序”翻译所得到的二进制代码称为目标程序。该过程称为“编译”。目标程序的扩展名为“.obj”,目标代码是机器指令,但是还不能运行。因为,目标程序还没有解决函数的调用问题,需要将各个目标程序与库函数进行连接。在对源程序的编译过程中,可能会发现程序中的语法错误。这时,需要重新利用编辑程序来修改源程序,然后再重新编译。

- 连接 经过编译后生成的目标文件是不能直接执行的,它需要经过连接之后才能生成可执行的代码。在DOS系统下,连接后所得到的可执行文件都是以“.exe”为其文件扩展名。

- 执行 经过编辑、编译、目标程序与库函数连接,最后生成完整的、可在操作系统下独立执行的程序。可执行程序的扩展名为“.exe”。这样,就可以直接在DOS/Windows环境下运行了。

开发一个C语言程序的基本步骤可用图1-1描述。C源程序经过编辑、编译、连接生