



中等职业教育“十一五”规划教材  
中职中专计算机类教材系列

# C 语言程序 设计与实训

宋世发 郑 泳 主编

中等职业教育“十一五”规划教材

中职中专计算机类教材系列

# C 语言程序设计与实训

宋世发 郑 泳 主 编

蔡其能 黄 敏 副主编

科 学 出 版 社

北 京

## 内 容 简 介

本书共分 11 章, 内容有: C 语言概述、输入输出的应用、C 语言的基本数据类型、运算符和表达式、语句及条件控制、循环控制、数组、函数、指针、结构体及文件等。本书突出“语言和程序设计”两大主题, 通过对本书的学习, 使学生能在 Turbo C 2.0 的环境下, 应用 C 语言进行初步的程序设计。

本书紧抓中职教育的特点, 用较大的篇幅安排了模仿试验、验证试验及试试看、动手做一整套试验及实训环节。本书旨在让读者在学中做, 做中学, 培养其良好的应用能力。

本书可作为中等职业技术学校计算机类专业的教材。

### 图书在版编目 (CIP) 数据

C 语言程序设计与实训/宋世发, 郑泳主编. —北京: 科学出版社, 2008  
(中等职业教育“十一五”规划教材·中职中专计算机类教材系列)  
ISBN 978-7-03-022514-6

I. C… II. ①宋…②郑… III. C 语言—程序设计—专业学校—教材  
IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 103992 号

责任编辑: 韩 洁 / 责任校对: 耿 耘

责任印制: 吕春珉 / 封面设计: 耕者设计工作室

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

铭浩彩色印务有限公司印刷

科学出版社发行 各地新华书店经销

\*

2008 年 8 月第 一 版 开本: 787×1092 1/16  
2008 年 8 月第一次印刷 印张: 13 1/2  
印数: 1—4 000 字数: 302 000

定价: 21.00 元

(如有印装质量问题, 我社负责调换〈环伟〉)

销售部电话 010-62134988 编辑部电话 010-62138978-8203

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

## 编 委 会

主 任 于 泽

编 委 (按拼音排序)

陈丁君 方风波 高 津 耿 杰 黄文胜

纪伟娟 雷 军 李 静 娄 彤 史少飞

宋世发 谭 永 滕 广 王咏丽 杨 利

杨少艳 于 虹 袁红梅 张秀生 张学义

张裕生 赵国增 甄慕华 朱 镇 祝朝映

## 前 言

C 语言是一种优秀的结构化程序设计语言,许多著名的系统软件和应用软件都是用 C 语言编写的。由于它具有灵活性、结构化等特点,因此被各大学以及各级各类高中、中等职业学校作为首选的入门程序设计语言,并将其作为专业基础课来设置,其目的是为学习其他的程序设计语言奠定算法及编程的基础,并培养学生的基本程序思想及编程能力。全国计算机二级等级考试、全国计算机应用证书考试都规定设置了 Turbo C 2.0 环境下的“C 语言程序设计”考试项目。当今流行的 Java 语言也来源于 C 语言。这些足以说明 C 语言的重要性。

针对中职学生的特点,并考虑到 C 语言在计算机类、信息类专业课程以及计算机公共基础课程体系中的地位和作用,为了培养学生对程序设计的理解、熟悉基本算法、掌握 C 语言的语法规则、建立起程序设计的思想,并通过对 C 语言的学习,体验一种全新的思维方式,本书的编写力求体现以下几方面的特点。

### 1. 体现中职学生的心智特点,力求教材通俗易懂

考虑到中职学生的年龄特点以及心理、智力因素,本书在编写的描述语言上力求做到简洁直白、通俗易懂,概念和术语力求用通俗易懂的、深入浅出的语言来表述;其次在内容的编排上力求做到由浅入深、例题引导,在叙述方式上加强了程序设计的方法和 C 语言的运用;在控制结构的讲解中,尽量以框图的形式介绍算法的逻辑思维方式,并在程序的实现上给予了详细的注释及说明;且在实践能力的培养上采用了循序渐进,试验案例由表及里的方法及措施。力求使教材适合中职学生的年龄及文化层次的特点。

### 2. 体现中职学生的思维特点,力求内容适度够用

C 语言程序设计是中等职业技术学校的学生第一次系统学习计算机的高级语言。全新的思维方式是习惯了学历教育的中职学生学习思维模式的大敌。本书充分考虑到了学生的特点,以及 C 语言课程在专业课程体系中的地位和作用。因此在内容的选材上加强了对基本数据类型、控制结构等基础内容的介绍,其目的就是介绍基本的程序思想为主,着重讲解语法规则以及程序的控制结构,将位运算、指针、结构体等较为抽象、复杂的知识点能删就删,能减则减。力求教材体现中职学生的思维特点,也实现职业教育的知识适度、够用。

### 3. 体现中职职业教育的特点,力求培养学生的应用能力

职业教育的主要目的就在于培养受教育者的职业技能和方法,向受教育者传授科学技术知识与生产劳动经验,开发受教育者的智力,并将其潜在的生产力转化为现实的生产力。正是基于这一原则,本书力求做到紧扣基础、循序渐进、面向应用;在教材的



每一章,用较大的篇幅安排了模仿试验、验证试验及试试看、动手做一整套试验及实训环节。通过让学习者在学中做,做中学来培养其良好的应用能力。

本书讲解的内容包括:C语言概述、输入输出的应用、C语言的基本数据类型、运算符和表达式、语句及条件控制、循环控制、数组、函数、指针、结构体及文件,共11章。

本书的所有程序都在 Turbo C 2.0 上编译、运行通过。

宋世发任本书主编,其中宋世发编写第1、2、3、4章,蔡其能编写第5章、黄敏编写第6章、郑泳编写第7、8、9章、桂峰编写第10章、方亚晴编写第11章。宋世发、郑泳负责全书的统撰。

在编写的过程中,袁方、王科、胡荣等同志做了大量的校对及程序验证工作,在此向这些付出了辛勤劳动的同志一并表示感谢。

本书在编写的过程中得到了许多兄弟学校的大力支持与关心,在此向他们表示感谢。

本书在编写过程中也参考了许多文献及成果。对本书参考书籍的作者、因特网上信息的提供者及作者,在此表示深深的敬意及诚挚的感谢。

由于编者水平所限、时间仓促,书中难免有欠妥之处,敬请广大读者、专家批评指正。

# 目 录

|                                 |    |
|---------------------------------|----|
| 第 1 章 C 语言程序概述 .....            | 1  |
| 1.1 C 语言概述 .....                | 2  |
| 1.1.1 C 语言的发展 .....             | 2  |
| 1.1.2 C 语言的特点 .....             | 3  |
| 1.2 基本语法成分 .....                | 3  |
| 1.2.1 C 语言的字符集 .....            | 3  |
| 1.2.2 标识符 .....                 | 4  |
| 1.2.3 关键字 .....                 | 4  |
| 1.2.4 运算符 .....                 | 5  |
| 1.2.5 分隔符 .....                 | 5  |
| 1.2.6 常量 .....                  | 5  |
| 1.2.7 注释符 .....                 | 5  |
| 1.3 C 语言程序结构 .....              | 6  |
| 1.3.1 简单的 C 语言程序介绍 .....        | 6  |
| 1.3.2 C 源程序的结构特点 .....          | 7  |
| 1.3.3 书写程序时应遵循的规则 .....         | 8  |
| 1.4 C 语言的编辑、编译和运行 .....         | 8  |
| 本章小结 .....                      | 9  |
| 思考与练习 .....                     | 10 |
| 实训一 C 语言的运行环境和运行过程 .....        | 10 |
| 第 2 章 输入、输出的应用 .....            | 15 |
| 2.1 数据输入/输出的概念及在 C 语言中的实现 ..... | 16 |
| 2.2 printf 函数 .....             | 16 |
| 2.2.1 printf 函数的一般形式 .....      | 16 |
| 2.2.2 转义字符的使用 .....             | 17 |
| 2.2.3 printf 函数的调用 .....        | 17 |
| 2.2.4 格式字符串 .....               | 18 |
| 2.3 scanf 函数 .....              | 22 |
| 2.3.1 scanf 函数的一般形式 .....       | 22 |
| 2.3.2 scanf 函数的格式说明 .....       | 22 |
| 2.4 其他输入/输出函数 .....             | 24 |
| 2.4.1 putchar 函数 .....          | 24 |
| 2.4.2 getchar 函数 .....          | 25 |



|                                |           |
|--------------------------------|-----------|
| 2.4.3 puts 函数 .....            | 26        |
| 2.4.4 gets 函数 .....            | 26        |
| 本章小结 .....                     | 27        |
| 思考与练习 .....                    | 27        |
| 实训二 输入与输出语句的使用 .....           | 29        |
| <b>第3章 C 语言的基本数据类型</b> .....   | <b>31</b> |
| 3.1 C 语言数据概述 .....             | 32        |
| 3.2 常量与变量 .....                | 33        |
| 3.2.1 常量和符号常量 .....            | 33        |
| 3.2.2 变量 .....                 | 34        |
| 3.3 整型数据 .....                 | 35        |
| 3.3.1 整型常量的表示方法 .....          | 35        |
| 3.3.2 整型变量 .....               | 36        |
| 3.4 实型数据 .....                 | 38        |
| 3.4.1 实型常量的表示方法 .....          | 38        |
| 3.4.2 实型变量 .....               | 39        |
| 3.5 字符型数据 .....                | 41        |
| 3.5.1 字符常量 .....               | 41        |
| 3.5.2 转义字符 .....               | 41        |
| 3.5.3 字符变量 .....               | 41        |
| 3.5.4 字符数据在内存中的存储形式及使用方法 ..... | 41        |
| 3.5.5 字符串常量 .....              | 42        |
| 3.6 变量赋初值 .....                | 43        |
| 3.7 各类数值型数据之间的混合运算 .....       | 44        |
| 本章小结 .....                     | 45        |
| 思考与练习 .....                    | 46        |
| 实训三 数据类型的理解及使用 .....           | 47        |
| <b>第4章 运算符与表达式</b> .....       | <b>50</b> |
| 4.1 基本概念 .....                 | 51        |
| 4.2 基本运算符和表达式 .....            | 53        |
| 4.2.1 算术运算 .....               | 53        |
| 4.2.2 赋值运算 .....               | 54        |
| 4.2.3 强制类型转换运算 .....           | 55        |
| 4.2.4 自增、自减运算 .....            | 56        |
| 4.2.5 关系运算 .....               | 57        |
| 4.2.6 逻辑运算 .....               | 58        |
| 4.2.7 逗号运算 .....               | 59        |
| 4.2.8 条件运算 .....               | 60        |

|                             |     |
|-----------------------------|-----|
| 本章小结 .....                  | 61  |
| 思考与练习 .....                 | 61  |
| 实训四  运算符与表达式 .....          | 62  |
| <b>第 5 章  语句及条件控制</b> ..... | 66  |
| 5.1  C 语句介绍 .....           | 67  |
| 5.2  赋值语句 .....             | 68  |
| 5.3  选择控制语句 .....           | 69  |
| 5.4  switch 语句 .....        | 74  |
| 5.4.1  switch 语句的一般格式 ..... | 74  |
| 5.4.2  switch 语句的执行过程 ..... | 74  |
| 5.5  条件控制应用举例 .....         | 76  |
| 本章小结 .....                  | 79  |
| 思考与练习 .....                 | 80  |
| 实训五  控制语句的使用 .....          | 82  |
| <b>第 6 章  循环控制</b> .....    | 85  |
| 6.1  循环语句介绍 .....           | 86  |
| 6.2  while 循环语句 .....       | 86  |
| 6.3  do-while 循环语句 .....    | 90  |
| 6.4  for 循环语句 .....         | 93  |
| 6.5  循环体中的控制语句 .....        | 98  |
| 6.5.1  break 语句 .....       | 98  |
| 6.5.2  continue 语句 .....    | 99  |
| 6.6  多重循环 (循环的嵌套) .....     | 101 |
| 6.7  循环控制应用举例 .....         | 104 |
| 本章小结 .....                  | 107 |
| 思考与练习 .....                 | 108 |
| 实训六 .....                   | 113 |
| 循环程序设计 (一) .....            | 113 |
| 循环程序设计 (二) .....            | 115 |
| <b>第 7 章  数组</b> .....      | 118 |
| 7.1  数组的概念 .....            | 119 |
| 7.2  一维数组的定义和引用 .....       | 120 |
| 7.2.1  一维数组的定义方式 .....      | 120 |
| 7.2.2  一维数组元素的引用 .....      | 121 |
| 7.2.3  一维数组的初始化 .....       | 123 |
| 7.2.4  一维数组程序举例 .....       | 124 |
| 7.3  字符数组 .....             | 127 |
| 7.3.1  字符数组的定义、初始化及引用 ..... | 127 |



|                     |            |
|---------------------|------------|
| 7.3.2 字符串处理函数       | 129        |
| 7.3.3 字符数组应用举例      | 133        |
| 本章小结                | 134        |
| 思考与练习               | 135        |
| 实训七 数组的应用           | 137        |
| <b>第8章 函数与编译预处理</b> | <b>139</b> |
| 8.1 函数基本概念          | 140        |
| 8.1.1 函数概述          | 140        |
| 8.1.2 函数的分类         | 141        |
| 8.2 函数的定义与调用        | 141        |
| 8.2.1 函数的定义         | 141        |
| 8.2.2 函数的调用         | 143        |
| 8.2.3 函数声明          | 144        |
| 8.3 函数间的参数传递        | 145        |
| 8.4 数组作为函数参数        | 147        |
| 8.5 变量的存储类别、作用域、生存期 | 149        |
| 8.5.1 作用域的概念        | 149        |
| 8.5.2 局部变量和全局变量     | 150        |
| 8.5.3 静态变量          | 152        |
| 8.6 编译预处理           | 153        |
| 8.6.1 宏定义           | 153        |
| 8.6.2 文件包含          | 156        |
| 本章小结                | 157        |
| 思考与练习               | 157        |
| 实训八 函数及宏定义的应用       | 159        |
| <b>第9章 指针</b>       | <b>161</b> |
| 9.1 地址和指针的概念        | 162        |
| 9.2 变量的指针和指向变量的指针变量 | 163        |
| 9.2.1 指针变量的定义       | 163        |
| 9.2.2 指针变量的引用       | 164        |
| 9.2.3 指针变量的运算       | 166        |
| 9.2.4 指针变量作为函数参数    | 167        |
| 9.3 数组的指针和指向数组的指针变量 | 168        |
| 9.3.1 指向数组元素的指针     | 169        |
| 9.3.2 通过指针指向数组元素    | 169        |
| 本章小结                | 172        |
| 思考与练习               | 172        |
| 实训九 指针的使用           | 176        |

|                                            |     |
|--------------------------------------------|-----|
| 第 10 章 结构体 .....                           | 178 |
| 10.1 结构体的定义和引用 .....                       | 179 |
| 10.1.1 结构体类型的定义格式 .....                    | 179 |
| 10.1.2 结构体变量的定义和初始化 .....                  | 180 |
| 10.1.3 结构体变量的引用 .....                      | 182 |
| 10.2 结构体数组 .....                           | 183 |
| 10.2.1 结构体数组的定义 .....                      | 183 |
| 10.2.2 结构体数组的引用 .....                      | 183 |
| 10.2.3 结构体数组的应用举例 .....                    | 184 |
| 10.3 链表 .....                              | 184 |
| 10.3.1 链表概述 .....                          | 184 |
| 10.3.2 处理链表所需的函数 .....                     | 185 |
| 10.3.3 链表的操作 .....                         | 186 |
| 本章小结 .....                                 | 188 |
| 思考与练习 .....                                | 189 |
| 实训十 结构体的应用 .....                           | 190 |
| 第 11 章 文件 .....                            | 192 |
| 11.1 文件概述 .....                            | 193 |
| 11.2 文件类型指针 .....                          | 194 |
| 11.3 文件的打开和关闭 .....                        | 194 |
| 11.3.1 文件的打开 (fopen 函数) .....              | 194 |
| 11.3.2 文件的关闭 (fclose 函数) .....             | 196 |
| 11.4 文件的读写 .....                           | 196 |
| 11.4.1 读写一个字符的函数——fgetc() 和 fputc() .....  | 196 |
| 11.4.2 读写数据块的函数——fread() 和 fwrite() .....  | 197 |
| 11.4.3 格式化读写函数——fscanf() 和 fprintf() ..... | 198 |
| 11.4.4 其他读写函数 .....                        | 198 |
| 11.5 文件的定位与出错检测 .....                      | 199 |
| 11.5.1 文件的定位 .....                         | 199 |
| 11.5.2 出错的检测 .....                         | 200 |
| 11.6 文件输入输出小结 .....                        | 200 |
| 本章小结 .....                                 | 201 |
| 思考与练习 .....                                | 201 |
| 实训十一 文件的应用 .....                           | 202 |
| 主要参考文献 .....                               | 204 |

# 第1章

## C 语言程序概述

### 知识目标

- C 语言的历史和发展过程。
- C 语言的特点、C 语言的基本语法成分、熟悉 C 程序的组成。
- C 程序的编译及链接过程、熟悉 C 程序的编译环境。

### 技能目标

- 熟悉 C 语言的语法成分，包括 C 语言的字符集、标识符、关键字、运算符、分隔符、常量、注释等，能正确地使用 C 语言的字符集、掌握 C 语言标识符的命名规则，能运用 C 语言的标识符命名变量，掌握 C 语言的注释方式和方法。
- 熟悉 C 程序的结构，掌握 C 程序的基本组成以及组成 C 程序的一些基本元素，如：预编译、函数体、声明语句、执行语句、main()函数等。
- 熟悉 C 语言的编辑、编译和运行环境，掌握 C 程序的实现过程。



本章介绍程序员编制程序所应该经历的步骤,你将了解到为什么要设计出 Turbo C 程序设计语言,并且简单地了解一下 C 语言的前景。本章向你展示如何开启编译程序,如何输入编辑和编译程序。尽管在这章中你会看到 C 代码,但你仍不会理解 C 代码。学习 C 语言之前,你应该学会如何输入、编译和运行 C 程序,只要你掌握了 Turbo C 的编译环境,你就可以在下章中开始学习 C 语言。

本章是 C 语言程序设计的入门部分,从整体上介绍 C 语言的起源和发展,讲述 C 语言的特点、结构和基础语法要点。

## 1.1 C 语言概述

### 1.1.1 C 语言的发展

C 语言是世界上广泛流行的程序设计语言之一,它适合于作系统描述语言,即用来写系统软件,也可以用来写应用软件。

早期的操作系统等系统软件主要是用汇编语言编写的(包括 UNIX 操作系统在内)。但是汇编语言存在明显的缺点,即可读性和可移植性都比较差,为了提高可读性和可移植性,最好改用高级语言,但是一般高级语言难以实现汇编语言的某些功能(汇编语言可以直接对硬件进行操作,例如对内存地址的操作、位操作等)。人们希望能找到一种既具有一般高级语言特性,又具有低级语言底层操作能力的语言,集它们的优点于一身,于是 C 语言在 20 世纪 70 年代初应运而生了。1978 年由美国电话电报公司(AT&T)的贝尔实验室正式发表了 C 语言,同时由 B.W.Kernighan 和 D.M.Ritchie 合著了影响深远的“THE C PROGRAMMING LANGUAGE”一书,通常简称为《K&R》,也有人称之为《K&R》标准。但是,在《K&R》中并没有定义一个完整的标准 C 语言,许多开发机构推出了自己的 C 语言版本,这些版本之间的微小差别不时引起兼容性上的问题,后来由美国国家标准学会 ANSI (American National Standard Institute) 在各种 C 语言版本的基础上制定了一个 C 语言标准,于 1983 年发表,通常称之为 ANSI C。1987 年 ANSI 又公布了新标准——89 ANSI C。目前广泛流行的各种 C 编译系统都是以它为基础的。

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 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之一。

在 C 的基础上,1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 进一步扩充和完善了 C 语言,成为一种面向对象的程序设计语言。C++ 提出了一些更为深入的概念,它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性,掌握起来有一定难度。但是, C 是 C++ 的基础, C++ 语言和 C 语言在很多方面是兼容的。因此,掌握了 C 语言,再进一步学习 C++ 就能以一种熟悉的语法来学习

面向对象的语言,从而达到事半功倍的目的。

目前最流行的C语言有以下几种:

- 1) Microsoft C 或称 MS C。
- 2) Borland Turbo C 或称 Turbo C。
- 3) AT&T C。

这些C语言版本不仅实现了ANSI C标准,而且在此基础上各自做了一些扩充,使之更加方便、完美。这些不同版本C语言之间有一定的差别,但对初学者来说,不必过多理会它们的差别,重在理解C语言的特点和编程方法。本书的叙述以TURBO C为准。

### 1.1.2 C语言的特点

C语言是一种通用、灵活、结构化、标准化、使用广泛的编程语言,能完成用户的各种任务,特别适合进行系统程序设计和对硬件进行操作的场合。C语言本身不对程序员施加过多限制,是一种专业程序员优先选择的语言。它有如下主要特点:

- 1) C语言简洁、紧凑,使用方便、灵活。ANSI C一共只有32个关键字。
- 2) 运算符丰富,共有34种。C语言把括号、赋值、逗号等都作为运算符处理,从而使C语言的运算类型极为丰富,可以实现其他高级语言难以实现的运算。
- 3) 数据结构类型丰富。
- 4) 具有结构化的控制语句。
- 5) 语法限制不太严格,程序设计自由度大。
- 6) C语言允许直接访问物理地址,能进行位(bit)操作,能实现汇编语言的大部分功能,可以直接对硬件进行操作。因此有人把它称为中级语言。
- 7) 生成目标代码质量高,程序执行效率高。
- 8) 与汇编语言相比,用C语言写的程序可移植性好。

C语言对程序员要求不高,程序员用C写程序会感到限制少、灵活性大,功能强,但较其他高级语言在学习上要困难一些。

## 1.2 基本语法成分

本节介绍C语言的字符集、关键字、标识符、运算符、分隔符和注释符等基本语法成分。

### 1.2.1 C语言的字符集

字符是组成语言最基本的元素。C语言字符集由字母、数字、空格、标点和特殊字符组成。在字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

- 1) 字母:小写字母a~z共26个,大写字母A~Z共26个。
- 2) 数字:0~9共10个。



3) 空白符: 空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时, 只起间隔作用, 编译程序对它们忽略。因此在程序中使用空白符与否, 对程序的编译不发生影响, 但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

4) 标点和特殊字符: 主要有 ! # % ^ & + - \* / = ~ < > \ | . , ; ? ' " ( ) { } [ ] 等。

由字符集中的字符可以构成 C 语言进一步的语法成分, 如标识符、关键字、特殊的运算符等。

## 1.2.2 标识符

在程序中使用的变量名、函数名、标号等统称为标识符, 用来标识各种程序成分。除库函数的函数名由系统定义外, 其余都由用户自定义。C 语言规定, 标识符只能是由字母(A~Z, a~z)、数字(0~9)、下划线(\_)组成的字符串, 并且其第一个字符必须是字母或下划线。

以下标识符是合法的:

a x x3 BOOK1 sum5 num\_1

以下标识符是非法的:

3s 以数字开头

s\*T 出现非法字符\*

-3x 以减号开头

bowy-1 出现非法字符- (减号)

在使用标识符时还必须注意以下几点:

标准 C 不限制标识符的长度, 但它受各种版本的 C 语言编译系统限制, 同时也受到具体机器的限制。例如在某版本 C 中规定标识符前八位有效, 当两个标识符前八位相同时, 则被认为是同一个标识符。Turbo C 中标识符最大长度为 32 个字符。

在标识符中, 大小写是有区别的。例如 BOOK 和 book 是两个不同的标识符。习惯上符号常量用大写字母表示, 而变量名等用小写字母表示。

标识符虽然可由程序员随意定义, 但不能与关键字同名, 也不能与系统预先定义的标准标识符(如标准函数)同名。标识符是用于标识某个量的符号, 因此, 命名应尽量有相应的意义, 以便阅读理解, 做到“见名知义”。

## 1.2.3 关键字

关键字是由 C 语言规定的具有特定意义的字符串, 通常也称为保留字。如类型说明符 int、double 等; 语句特征 if、switch、while 等; 预处理命令 include、define 等。关键字是构成 C 语言的语法基础, 用户定义的标识符不应与关键字相同, 也不能对关键字进行重新定义, 如表 1.1 所示。

表 1.1 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   | static | sizeof   | struct | switch   | typedef  | union   |
| unsigned | void   | volatile | while  |          |          |         |

9 种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。

Turbo C 扩充了 11 个关键字：

asm    \_cs    \_ds    \_es    \_ss    cdecl    far  
huge    interrupt    near    pascal



在 C 语言中，关键字都是小写的。

## 1.2.4 运算符

C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。根据参加运算对象的个数，运算符可分为单目运算符、双目运算符和三目运算符。

## 1.2.5 分隔符

C 语言中的分隔符有逗号和空白两种，逗号主要用在类型说明和函数参数表中，分隔各个变量。空白包括：空格符、制表符、换行符，其多用于语句各单词之间，作间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔，否则将会出现语法错误。例如把“int a;”，写成“inta;”，C 编译器会把“inta”当成一个标识符处理，其结果必然出错。

## 1.2.6 常量

C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种，在后面章节中将专门给予介绍。

## 1.2.7 注释符

注释符是以“/\*”开头并以“\*/”结尾的字符串。在“/\*”和“\*/”之间的即为注释。程序编译时，不对注释做任何处理。注释可出现在程序中的任何位置，其用来向用户提示或解释程序的意义。在调试程序时对暂不使用的语句也可用注释符括起来，使编译跳过不做处理，待调试结束后再去掉注释符。

## 1.3 C 语言程序结构

### 1.3.1 简单的 C 语言程序介绍

为了说明 C 语言源程序结构的特点,先看以下几个程序。这几个程序由简到难,表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍,但可从这些例子中了解到组成一个 C 源程序的基本部分和书写格式。

**【例 1.1】** 打印“Hello world!”。

```
main()
{
    printf("Hello world! \n");
}
```

其中, main 是主函数的函数名,表示这是一个主函数。

每一个 C 源程序都必须有,且只能有一个主函数(main 函数)。

函数调用语句, printf 函数的功能是把要输出的内容送到显示器去显示。

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。

双引号内字符串原样输出,“\n”是回车换行符。

**【例 1.2】** 显示正弦函数的运算结果。

```
#include<math.h>          /* include 称为文件包含命令*/
#include<stdio.h>          /*扩展名为.h的文件称为头文件*/
main()
{
    double x,s;             /*定义两个实数变量,以被后面程序使用*/
    printf("input number:\n"); /*显示提示信息*/
    scanf("%lf",&x);         /*从键盘获得一个实数 x*/
    s=sin(x);               /*求 x 的正弦,并把它赋给变量 s*/
    printf("sine of %lf is %lf\n",x,s); /*显示程序运算结果*/
}                            /* main 函数结束*/
```

程序的功能是从键盘输入一个数 x,求 x 的正弦值,然后输出结果。在 main()之前的两行称为预处理命令(详见后面)。预处理命令还有其他几种,这里的 include 称为文件包含命令,其意义是把尖括号<>或引号""内指定的文件包含到本程序来,成为本程序的一部分。被包含的文件通常是由系统提供的,其扩展名为.h,因此也称为头文件或首部文件。C 语言的头文件中包括了各个标准库函数的函数原型。因此,凡是在程序中调用一个库函数时,都必须包含该函数原型所在的头文件。在本例中,使用了三个库函数:输入函数 scanf,正弦函数 sin,输出函数 printf。sin 函数是数学函数,其头文件为 math.h 文件,因此在程序的主函数前用 include 命令包含了 math.h。scanf 和 printf 是标准输入输出函数,其头文件为 stdio.h,在主函数前也用 include 命令包含了 stdio.h 文件。