



普通高等教育“十二五”规划教材 计算机系列
中国科学院教材建设专家委员会“十二五”规划教材

程序设计基础教程

CHENGXU SHEJI JICHU JIAOCHENG
(C语言版)

黄思先 刘必雄 主 编



科学出版社

普通高等教育“十二五”规划教材·计算机系列
中国科学院教材建设专家委员会“十二五”规划教材

程序设计基础教程（C 语言版）

黄思先 刘必雄 主 编

刘玥波 张素莉 副主编

陈梅雯 刘秀玲
李丽珊 参 编
林 娟 郑珂晖

科 学 出 版 社

北 京

内 容 简 介

本书以 C 语言为依托, 结合典型实例, 全面、系统地介绍了程序设计的思想和方法。采用案例引入和问题驱动的方式, 以大量的实例, 深入浅出地介绍了 C 语言的基础知识、运算符与表达式、程序的控制结构、构造数据类型、函数、指针和文件等内容。为了提高读者的实际编程能力, 每章配有多套“上机练练练”示范题, 并配有习题, 以利于读者巩固所学的知识, 提高自学能力。

本书可作为高等院校本科生计算机及相关专业“程序设计基础”、“高级语言程序设计”、“C 语言程序设计”等课程的教材, 也可作为非计算机专业学生的计算机程序设计课程的教材。

图书在版编目 (CIP) 数据

程序设计基础教程: C 语言版/黄思先, 刘必雄主编. —北京: 科学出版社, 2011
ISBN 978-7-03-032894-6

I. ①程… II. ①黄… ②刘… III. ①C 语言—程序设计—高等学校—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字 (2011) 第 244477 号

责任编辑: 戴 薇 郭丽娜/责任校对: 王万红

责任印制: 吕春珉/封面设计: 东方人华平面设计部

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>

新科印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2012 年 1 月第 一 版 开本: 787×1092 1/16

2012 年 1 月第一次印刷 印张: 18 1/2

字数: 432 000

定价: 33.00 元

(如有印装质量问题, 我社负责调换<新科>)

销售部电话 010-62142126 编辑部电话 010-62134021

版权所有, 侵权必究

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

前 言

程序设计基础课程作为高等院校学生学习计算机程序设计的入门课程,对培养学生程序设计的基本方法和技巧,以及运用编程解决相关专业领域问题的能力具有重要作用。

本书根据教育部高等院校计算机科学与技术教学指导委员会编制的《关于进一步加强高等学校计算机基础教学的意见暨计算机基础课程教学基本要求(试行)》,以“提高学生实践能力,培养学生的编程能力”为宗旨,结合一线教师的多年教学实践经验编写而成。

本书以“学语言练程序设计”为目标,以标准 C 为框架,以 Microsoft Visual C++ 6.0 为编程环境,详细介绍了 C 语言程序设计的基本规范、思路和方法。针对高等院校学生的特点,采用案例引入和问题驱动的方式,运用通俗易懂的文字,循序渐进、由浅入深地介绍了 C 语言程序设计,力求做到符合学习规律。

为了在实验课程中切实培养学生的实际动手编程能力,本书配有多套“上机练练练”示范题,这也是本书的一个特色。每套“上机练练练”由“教师示范题”、“学生模拟题”和“拓展题”共三个部分组成。其中“学生模拟题”是在“教师示范题”基础上简单的扩展;而“拓展题”则是“教师示范题”和“学生模拟题”从题目难度及规模上进一步的拓展,需要独立思考和综合分析才能完成解题。本书为“上机练练练”的每个实验题目提供了测试数据,可以通过“程序评测系统”进行评测,并在每次实验课中对学生制作的程序进行现场评分,所有的测试数据及相关程序可登录网站 <http://www.abook.cn> 下载。

本书根据作者多年的教学实践编写而成,是集理论学习、实验、习题于一体的教学书籍。全书分为 7 章,第 1 章介绍了 C 语言的发展背景、C 程序的基本组成要素与 C 程序的结构、数据类型、常量、变量、数据的输入/输出等内容;第 2 章介绍了运算符及表达式;第 3 章介绍了 C 语言进行结构化程序设计的基本方法,包括顺序结构、选择结构、循环结构;第 4 章介绍了数组的定义、数组元素的引用以及数组应用的一些基本算法,并对结构体、共用体及枚举类型作了较详细的介绍;第 5 章介绍了函数以及编译预处理;第 6 章充分阐述了 C 语言的指针;第 7 章对 C 语言文件操作进行了较详细的阐述。另外,每章都配有“上机练练练”,并附有大量的习题。

本书由黄思先、刘必雄担任主编。编写分工如下:第 1 章由李丽珊编写;第 2 章由刘秀玲编写;第 3 章由林娟编写;第 4 章由刘必雄编写;第 5 章由陈梅雯编写;第 6 章由黄思先编写;第 7 章由郑珂晖编写;张素莉和刘玥波也参与了本书的编写工作。黄思先对全书的结构和编写作了统筹策划,并提出了“上机练练练”示范题的架构;刘必雄负责全书的统稿和修改。

在组织和编写本书过程中,得到同行以及科学出版社相关同志的热情鼓励 and 大力支持,在此谨向他们以及关心和支持本书编写工作的各方面人士表示衷心的感谢!

由于编者水平有限,书中难免有错误和不妥之处,恳请专家和广大读者批评指正,以便再版时进一步完善。

目 录

前言

第 1 章 C 语言基础知识	1
1.1 C 语言概述	1
1.1.1 C 语言的由来与标准化	1
1.1.2 C 语言的特点和应用	1
1.2 C 程序的结构	2
1.2.1 C 程序的基本结构	2
1.2.2 简单 C 程序分析	3
1.2.3 C 程序的具体构成	5
1.2.4 C 程序的上机执行过程	8
1.2.5 Visual C++ 6.0 的上机步骤	9
1.3 C 语言的数据类型	10
1.3.1 基本数据类型	11
1.3.2 变量	11
1.3.3 常量	13
1.3.4 数据在计算机内的存储	17
1.4 C 程序的输入/输出	19
1.4.1 用函数实现输入/输出	19
1.4.2 用对象流实现输入/输出	24
小结	27
上机练练练	27
习题	30
第 2 章 运算符与表达式	32
2.1 概述	32
2.2 算术运算符与算术表达式	33
2.2.1 算术运算符	33
2.2.2 算术表达式	33
2.2.3 自增和自减运算符	34
2.3 赋值运算符与赋值表达式	35
2.3.1 赋值运算符	35
2.3.2 复合赋值运算符	36
2.3.3 赋值表达式	36
2.4 关系运算符与关系表达式	37
2.4.1 关系运算符	37
2.4.2 关系表达式	38
2.5 逻辑运算符与逻辑表达式	38
2.5.1 逻辑运算符	38

2.5.2 逻辑表达式	39
2.6 位运算符与位运算表达式	40
2.6.1 位运算符	40
2.6.2 位运算表达式	41
2.7 条件运算符与条件表达式	46
2.8 逗号运算符和逗号表达式	47
2.9 求字节运算符与求字节表达式	47
2.10 数据类型转换	48
2.10.1 自动类型转换	48
2.10.2 强制类型转换	51
小结	51
上机练练练	51
习题	55
第 3 章 程序的控制结构	60
3.1 结构化程序设计概述	60
3.2 顺序结构	61
3.2.1 C 语言的语句	61
3.2.2 顺序结构应用举例	62
3.3 选择结构	63
3.3.1 if 语句	63
3.3.2 switch 语句	68
3.4 循环结构	70
3.4.1 while 语句	70
3.4.2 do-while 语句	71
3.4.3 for 语句	72
3.4.4 循环嵌套	74
3.5 其他控制语句	76
3.5.1 break 语句	76
3.5.2 continue 语句	77
3.5.3 goto 语句	79
小结	79
上机练练练	80
习题	85
第 4 章 构造数据类型	90
4.1 数组	90
4.1.1 数组概述	90
4.1.2 一维数组	92
4.1.3 二维数组	105
4.1.4 字符数组和字符串	113

4.2 结构体	120
4.2.1 结构体类型声明	121
4.2.2 结构体变量	122
4.2.3 结构体数组	126
4.2.4 位段	129
4.2.5 结构体的应用举例	131
4.3 共用体	132
4.3.1 共用体类型声明	132
4.3.2 共用体变量	133
4.3.3 共用体的应用举例	135
4.4 枚举类型	136
4.4.1 枚举类型声明	136
4.4.2 枚举变量	137
4.4.3 枚举类型的应用举例	139
4.5 自定义类型名	140
小结	141
上机练练练	141
习题	152
第5章 函数	159
5.1 函数概述	159
5.2 函数的定义和调用	161
5.2.1 函数的定义	161
5.2.2 函数的调用	164
5.3 函数的参数传递	166
5.3.1 值传递	166
5.3.2 地址传递	168
5.4 函数的调用方式	172
5.4.1 函数的嵌套调用	172
5.4.2 函数的递归调用	174
5.5 变量的作用域和生存期	178
5.5.1 变量的作用域	178
5.5.2 变量的生存期和存储类型	180
5.6 宏定义	186
5.6.1 无参宏定义	186
5.6.2 带参宏定义	187
小结	190
上机练练练	190
习题	194
第6章 指针	200
6.1 指针与指针变量	200
6.1.1 指针的基本概念	200

6.1.2	指针变量的定义	201
6.1.3	指针变量的赋值	202
6.1.4	指针运算符	203
6.2	指针与数组	204
6.2.1	指针与一维数组	204
6.2.2	移动指针及两指针相减运算	207
6.2.3	指针比较	208
6.2.4	指向字符串的指针	209
6.2.5	指针与二维数组	212
6.3	指向指针的指针	220
6.3.1	指向指针的指针的概念	220
6.3.2	指向指针的指针变量定义	220
6.3.3	指向指针的指针变量的应用	221
6.4	指针与结构体	222
6.4.1	指向结构体变量的指针变量	222
6.4.2	指向结构体数组的指针变量	223
6.4.3	通过指针变量存取位段数据	224
6.5	指向共用体和枚举型的指针	225
6.5.1	指向共用体变量的指针变量	225
6.5.2	指向枚举型的指针变量	226
6.6	函数与指针	226
6.6.1	返回指针的函数	226
6.6.2	指针变量作函数的形参	228
6.6.3	主函数 main() 的形参	230
6.6.4	指向函数的指针	231
6.7	链表	234
6.7.1	单向链表的建立	234
6.7.2	单向链表的查找操作	235
6.7.3	单向链表结点的插入操作	236
6.7.4	单向链表结点的删除操作	236
6.7.5	一个完整的单向链表应用程序	237
	小结	240
	上机练练练	241
	习题	248
第 7 章	文件	257
7.1	文件的基本概念	257
7.2	文件指针	258
7.3	文件的打开与关闭	259
7.3.1	文件打开函数	259
7.3.2	文件关闭函数	260

7.4 文件的读写	261
7.4.1 字符读写函数	261
7.4.2 数据块读写函数	265
7.4.3 数据格式化读取函数	267
7.4.4 其他的文件读写函数	268
7.5 文件的定位函数	270
7.6 文件出错检测函数	272
小结	273
上机练练练	273
习题	278
附录 A ASCII 码表	280
附录 B 运算符和结合性	281
附录 C C 语言常用库函数	282
参考文献	286

第 1 章 C 语言基础知识

C 语言是一种功能很强的高级计算机语言。本章主要介绍 C 语言的发展背景、C 程序的基本组成要素与 C 程序的结构、数据类型、常量、变量、数据的输入/输出等内容。

1.1 C 语言概述

1.1.1 C 语言的由来与标准化

C 语言是由美国贝尔研究所的 Dennis M. Ritchie 于 1972 年推出的。随着微型计算机的日益普及,出现了许多 C 语言版本,于是需要制定一套标准来规范不同版本的 C 语言。1978 年, Brian W. Kernighan 和 Dennis M. Ritchie (简称 K&R) 正式出版了著名的《The C Programming Language》一书,此书介绍的 C 语言成为后来广泛使用的 C 语言基础版本,被称为标准 C 语言。1978 后, C 语言已先后被移植到大、中、小及微型机上。

C 语言的标准化工作是从 20 世纪 80 年代初期开始的。1983 年,美国标准学会(American National Standards Institute, ANSI) 根据各种 C 语言版本对 C 语言进行了扩充和发展,颁布了 C 语言的新标准 ANSI C。

由于 C 语言的不断发展,1987 年美国标准学会在综合各种 C 语言版本的基础上,又颁布了新标准,为了与标准 ANSI C 有区别,称之为 87 ANSI C; 1990 年,国际标准化组织(International Organization for Standardization, ISO) 接受了 87 ANSI C 作为 ISO C 的标准。这是目前功能最完善、性能最优良的 C 语言新版本,当前流行的 C 编译系统都是以 87 ANSI C 为基础的。本书讲述的内容也是以 ANSI C 为基础的。

1.1.2 C 语言的特点和应用

1. C 语言的特点

1) 语言简洁、紧凑、灵活。C 语言一共只有 37 个关键字、9 种控制语句,程序书写形式自由,主要用小写字母表示。

2) 运算符和数据类型丰富。C 语言共有 34 种运算符。C 语言的数据类型有基本类型、构造类型、指针类型、空类型 4 大类,这 4 大类型结合能实现各种复杂的数据结构。

3) 面向过程的结构化程序设计语言。结构化语言的显著特点是代码、数据的模块化,即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰,便于使用、维护以及调试。C 语言是以函数为主体结构,函数间通过调用传递参数和返回结果,同时采用 3 种基本结构来控制程序流向,从而使程序完全结构化。

- 4) 生成目标代码质量高。一般只比汇编程序生成的目标代码效率低 10%~20%。
- 5) 语法限制不太严格。一般的高级语言语法检查比较严格,能够检查出几乎所有的语法错误,而 C 语言允许程序编写者有较大的自由度。
- 6) 可直接访问物理地址。C 语言利用指针操作就可以直接进行靠近硬件的操作,可以用来写系统软件。
- 7) 适用范围广。C 语言适合于多种操作系统,如 DOS、UNIX、Windows XP、Windows NT 等,也适用于多种机型。

2. C 语言的应用

作为现代计算机语言的代表之一,C 语言展现出强大的生命力。它可以作为工作系统设计语言,编写系统应用程序,也可以作为应用程序设计语言,编写不依赖于计算机硬件的应用程序。目前 C 语言广泛应用在单片机以及嵌入式系统的开发上。

1.2 C 程序的结构

1.2.1 C 程序的基本结构

C 程序一般由 4 部分构成,分别是预处理命令、语句、函数和注释。C 程序的基本结构如图 1-1 所示。

1) 预处理命令以“#”开头,末尾没有分号。C 提供 3 种预处理命令,即宏定义命令、文件包含命令及条件编译命令。

2) 全局声明定义语句、局部声明定义语句和执行语句,都是以分号“;”结尾的语句。它是 C 程序的基本组成单位。当声明定义语句的位置在函数之外时,它是全局声明定义语句,用于进行全局对象的声明,在其作用域内都可以被访问。当声明定义语句的位置在函数之内时,则是局部声明定义语句,它只在本函数范围内可访问。

3) 函数由函数头和函数体构成。函数体中又包含着局部声明定义语句和执行语句两部分。C 程序中的函数可以是一个或多个,但其中必须有且只有一个函数名为 `main` 的函数,称之为“主函数”。不论主函数位于程序的什么位置,C 程序都是从主函数开始执行,并结束于主函数。函数之间通过相互调用来执行。

图 1-1 中并没有体现注释,这是因为注释可以出现在程序的任意位置,但并不参与程序的运行,只是用于帮助读者阅读源程序。注释分为行注释和块注释两种。

1) “//”为行注释符号,只在本行内起作用,在其之后的内容为行注释内容。

2) “/*.....*/”为块注释符,其以“/*”开头,“*/”为结尾,这两者之间的内容即为注释内容,可跨多行。

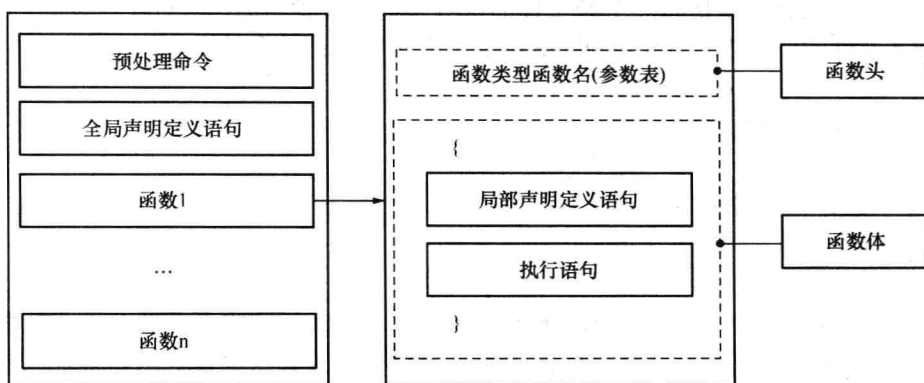


图 1-1 C 程序的基本结构

给程序加注释通常遵循以下原则：①一个程序至少应该有一条注释语句，对整个程序功能进行说明；②对程序中主要的变量应该加必要的注释，以说明变量的作用；③对程序中重要的语句块应该加必要的注释，以说明语句块的作用。

1.2.2 简单 C 程序分析

下面通过两个具体的例子来分析 C 程序的结构特点。

【例 1.1】输出“Hello World!”的程序。

程序代码：

```
#include <stdio.h>
//全局声明定义语句为空
int main()
{
    //局部声明定义语句为空
    printf("Hello World!\n");
    return 0;
}
```

运行结果：

```
Hello World!
```

程序说明：

例 1.1 是一个非常简单的程序，程序由一条预处理命令和一个 main() 函数组成。main() 函数体中也只有两条执行语句。

程序第一行是一条预处理命令：

```
#include <stdio.h>
```

它的作用是把尖括号中名为“stdio.h”的头文件包含到程序中。stdio.h 中包含了库函数 printf 的相关定义信息。语句“printf(“Hello World!\n”);”就是对库函数 printf 的调用，用于在屏幕上输出“Hello World!”。输出的字符序列中没有“\n”，它是一种转义字符，其作用是回车换行。

所谓“文件包含”是指将一个源文件的内容全部合并到当前源文件中,即将一个源文件包含到本文件中。C 语言提供#include 命令用来实现文件包含的操作。

【语法格式】

#include <文件名>

或

#include "文件名"

说明

1) 在编译预处理时,用指定的“包含文件名”中的文本内容替代该语句,使包含文件的全部内容成为本程序清单的一部分。在 C 语言程序设计中经常会用到系统函数,有时程序员自己也要定义宏、结构体类型、全局变量等,它们的声明往往都分门别类地放在不同的“头文件”(h)中,通过文件包含命令将它们包含到程序的清单中。

2) 文件包含的两种格式在使用时还是有一定的区别,第一种用“<>”括起来的格式,用来包含那些由系统提供的头文件存放在指定目录中(一般在 include 子目录中)。第二种用“””括起来的格式,用来包含那些由程序员自己定义的放在当前目录下的头文件。也就是说如果用“<>”的格式,系统就在 include 目录下查找头文件,如果用“””格式,系统首先在当前目录下查找,如果找不到再到 include 目录下查找头文件。为了避免寻找包含文件时出错,通常库函数使用“<>”格式,一般用户文件可使用“””格式。

注意

1) #include 命令只能包含一个头文件,如果想包含多个头文件,则必须用多条文件包含命令。

2) 使用文件包含后会使得编译后的目标文件变长,为使目标文件不至于过长,在定义包含文件时,其内容不宜过多。

3) 包含文件除了可以将系统函数和系统宏定义包含到用户程序中,还有一个很重要的功能,即是将多个源程序清单合并成一个源程序后进行编译。

程序的第二行是 main()函数的函数头,其函数类型是 int,函数名为 main,“()”是函数的标识,括号中没有内容,表明该函数无参数。也可书写为“int main(void)”。

main()函数的函数体内,还有语句“return 0;”,其用来结束 main()函数的执行,并将 0 值返回,把控制权还给操作系统。

【例 1.2】 用结构体变量存储 3 门课程成绩和总分,调用函数计算平均成绩并输出。

程序代码:

```
#include <stdio.h>
struct score //全局结构体类型 score 声明
{
    int score1, score2, score3;
    float sum;
};
struct score math={85,80,81}; //score 类型全局变量 math 定义
void summary()
{
```

```

    math.sum=math.score1+math.score2+math.score3;
    //全局变量 math 的访问
}
int main()
{
    float average;
    summary();                //summary 函数的调用
    average=math.sum/3.0;     //全局变量 math 的访问
    printf("Average Score is %f",average);
    return 0;
}

```

运行结果:

```
Average Score is 82.000000
```

程序说明:

例 1.2 明显比例 1.1 复杂多了,它包含一条预处理命令、一条全局结构体类型声明语句、一条全局结构体变量定义语句和两个函数的定义: `main()`和 `summary()`。但是归结起来,依旧是预处理命令、语句、函数这一基本结构。

```

struct score
{
    int score1,score2,score3;
    float sum;
};

```

它是一条语句,用于声明结构体名为 `score` 的结构体的结构,其有 4 个成员,即 `score1`、`score2`、`score3` 和 `sum`。

语句“`struct score math={85,80,81};`”是定义一个 `score` 结构体类型的全局变量 `math`,并为其中的 3 个成员 `score1`、`score2` 和 `score3` 分别赋初值为 85、80 和 81。

`summary()`函数的功能是访问全局变量 `math`,将其 3 个成员 `score1`、`score2` 和 `score3` 的总和计算后存储在 `math` 的 `sum` 成员中。

`main()`函数中定义了一个局部变量 `average`,用于存储全局变量 `math` 的 3 个 `score` 成员总和的平均值,并输出。

通过实例可以看到,由于全局变量 `math` 的定义位置是在所有函数之前的,所以 `summary()` 函数和 `main()`函数均可访问它。

1.2.3 C 程序的具体构成

任何程序设计语言如同自然语言一样,都具有一套对字符、单词及一些特定符号的使用规定,也有对语句、语法等方面的使用规则。在 C 语言中所涉及的规定很多,其中主要有基本字符集、标识符、关键字、语句和标准库函数等。这些规定构成了 C 程序的最小的语法单位。

1. 基本字符集

一个 C 程序是 C 语言基本字符构成的一个序列,C 语言的基本字符集由以下成员构成。

1) 字母: A~Z、a~z。

2) 数字: 0~9。

3) 空白符: 空格符、制表符、换行符等。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不计。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, 如表 1-1 所示。

表 1-1 标点与特殊字符

字 符	名 称	字 符	名 称	字 符	名 称
!	感 叹 号	+	加 号	"	引 号
#	井 号	=	等 号	{	左花括号
%	百 分 号	~	波 浪 号	}	右花括号
^	折 音 符	[	左方括号	,	逗 号
&	和 号	]	右方括号	.	句 号
*	星 号	'	撇 号	<	小 于 号
(	左 括 号		竖 线	>	大 于 号
_	下 画 线	/	反 斜 杠	/	除 号
)	右 括 号	;	分 号	?	问 号
-	连 字 符	:	冒 号		

基本字符集可以构成 C 程序的词汇, 具体分为标识符、关键字、运算符、分隔符、常量和注释符等 6 类。在字符常量、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。

2. 标识符

在程序中有许多需要命名的对象, 以便在程序的其他地方使用。如何表示在一些不同地方使用的同一个对象, 最基本的方式就是对对象命名, 通过名称在程序中建立定义与使用的关系, 建立不同使用之间的关系。为此, 每种程序语言都规定了在程序中描述名称的规则, 这些名称包括变量名、常数名、数组名、函数名、文件名、类型名等, 通常被统称为“标识符”。

C 语言规定, 标识符由字母、数字和下画线“_”组成, 它的第一个字符必须是字母或下画线。C 语言区分大小写, 标识符中同一个字母的大写与小写被看作是不同的字符。如表 1-2 所示给出了合法的和非法的两组 C 语言标识符。

表 1-2 合法的和非法的 C 语言标识符比较

合法的标识符	不合法的标识符	说 明
call_name	call..name	非字母数字或下画线组成的字符序列
test39	39test	非字母或下画线开头的字符序列
_string1	-string1	非字母或下画线开头的字符序列

在 C 程序中, 标识符的使用很多, 使用时要注意语言规则。下面给出标识符使用的一些建议。

1) 取名尽量直观, 做到见名知意, 便于记忆和阅读。

2) 可以在标识符中用下画线和大小写间隔来增强可读性, 如 studentname, 换成

StudentName 或 student_name 就直观多了。

3) 标准 C 语言不限制标识符的长度,但是各种版本的 C 语言编译系统和不同的机型会对其有限制,具体参考不同版本的 C 语言来使用。对于初学者而言,定义标识符长度最好不超过 8 个。

3. 关键字

C 语言有一些具有特定含义的关键字,用作专用的定义符,也称为保留字。这些特定的关键字不允许用户作为自定义的标识符使用。C 语言关键字绝大多数是由小写字母构成的字符序列。常用的 C 语言的关键字是 37 个,大致分为类型说明符、语句定义符、预处理命令字等。关键字具体如下:

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

4. 表达式

表达式由运算符、常量及变量构成。在表达式中的常量和变量必须遵循运算符的运算规则参与运算,并得出运算结果,该结果就是表达式的值。

5. 语句

语句是组成程序的基本单位,它能完成特定操作,语句的有机组合能实现指定的计算处理功能。所有程序设计语言都提供了满足编写程序要求的一系列语句,它们都有确定的形式和功能。C 语言中的语句有以下几类:①控制语句(if、switch、for、while、do-while、break、continue、return、goto);②表达式语句;③复合语句;④函数调用语句;⑤空语句;这些语句的形式和使用见后续相关章节。

6. 标准库函数

标准库函数不是 C 语言本身的组成部分,它是由 C 语言编译系统提供一些非常有用的功能函数。例如,C 语言没有输入/输出语句,也没有直接处理字符串的语句,而一般的 C 语言编译系统都提供了完成这些功能的函数,称为标准库函数。在 C 语言处理系统中,标准库函数存放在不同的头文件中,使用时只要把头文件包含在用户程序中,就可以直接调用相应的库函数了。

标准库函数是语言处理系统中一种重要的软件资源,在程序设计中充分利用这些函数,常常会收到事半功倍的效果。所以,读者在学习 C 语言本身的同时,应逐步了解和掌握标准库中各种常用函数的功能和用法,避免自行重复编写这些函数。

标准函数库中的函数名,可以用做标识符,重新赋予其新的含义,但是一般不建议挪作他用,如果挪作他用,将失去原先的意义。

需要说明的是，不同 C 语言编译系统提供的标准库函数在数量、种类、名称及使用上都有些差异。但就一般编译系统而言，常用的标准函数基本上是相同的（见附录 C）。

综上所述，可以对 C 程序的语言构成有个从整体到局部的层次划分，具体如图 1-2 所示。

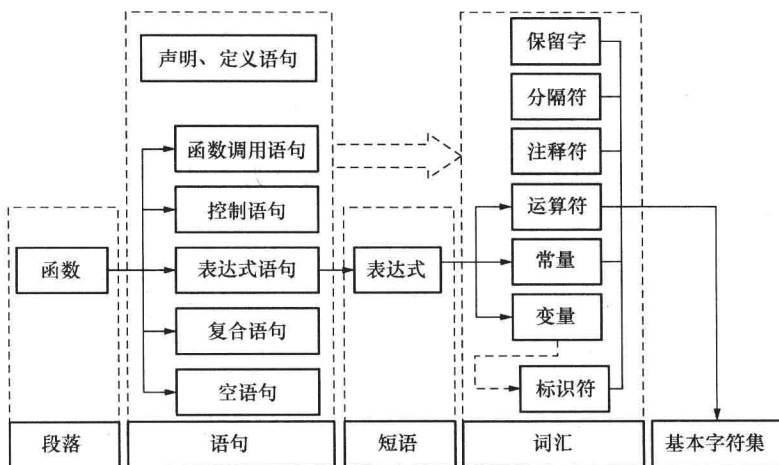


图 1-2 C 程序的语言构成

从图 1-2 中可以看到，C 程序的语言构成同人类语言构成非常类似。所不同的是这门语言是用于人类同计算机沟通联系的，是单向的。在这些构成成员中，各成员依据一定的准则连接为一个整体，如函数之间依靠相互调用联系起来，而语句之间则依靠顺序、选择和循环结构这 3 种基本结构连接起来。

1.2.4 C 程序的上机执行过程

1. 编辑

编辑是指用户把编写好的 C 语言源程序输入计算机，并以文本文件的形式存放在磁盘上。其标识为“文件名.c”，C++文件标识为“文件名.cpp”。

2. 编译

编译是指把 C 语言源程序翻译成用二进制指令表示的目标文件。编译过程由 C 语言编译系统提供的编译程序完成。编译程序自动对源程序进行句法和语法检查，当发现错误时，就将错误的类型和所在的位置显示出来，提供给用户，以帮助用户修改源程序中的错误。如果未发现句法或语法错误，就自动形成目标代码并对目标代码进行优化后生成目标文件。目标程序的文件标识是“文件名.obj”。这是系统规定的形式，扩展名“obj”是目标程序的文件类型标识。

3. 连接

尽管生成了目标程序，计算机还是不能执行的。程序连接过程是用系统提供的连接程序（也称链接程序或装配程序）将目标程序、库函数或其他目标程序连接装配成可执行的目标程序。可执行程序的文件名为“文件名.exe”，扩展名“exe”是可执行程序的文件类型标识。