

高等院校规划教材

PUTONG GAODENG
YUANXIAO
JISUANJI JICHU
JIAOYU XILIE JIAOCAI

普通高等院校计算机基础教育系列教材

C程序设计技术

主 编 刘慧君 熊 壮

CHENGXU SHEJI JISHU



重庆大学出版社

<http://www.cqup.com.cn>

高等院校规划教材

PUTONG GAODENG
YUANXIAO
JISUANJI JICHU
JIAOYU XILIE JIAOCAI

普通高等院校计算机基础教育系列教材

C程序设计技术

主 编 刘慧君 熊 壮

CHENGXU SHEJI JISHU

编 者 (按姓氏笔画排序)

王欣如 冉春林 伍 星

刘慧君 何 频 陈 策

张全和 吴漪菡 林望如

熊 壮

重庆大学出版社

内容提要

本书针对程序设计语言初学者,以C语言为载体,以微软 Visual C++6.0 为环境,通过讨论C程序设计的一般过程和方法,重点介绍了结构化程序设计的基本思想和实现方法。本书通过数据组织、控制结构、文件处理等程序设计基础知识的讨论,向读者讲授使用C语言进行程序设计的基本方法;通过对指针与函数关系、指针与数组关系、指针数组、动态数组实现方法、构造数据类型使用方法等方面的讨论,向读者讲授C语言特有的一些重要知识,使读者能够循序渐进地掌握使用C语言开发各类常见应用程序的基本技能。

本书在附录中提供了ASCII码表、C程序设计中常用的标准库函数、使用 Visual C++ 6.0 集成环境开发C程序的基本方法等重要学习资料。

本书覆盖了C语言的应用基础,内容深入浅出、语言流畅、例题丰富,适合作为程序设计语言课程初学者的教材,对于程序设计爱好者也是极佳的入门教材或参考书。

图书在版编目(CIP)数据

C 程序设计技术/刘慧君,熊壮主编.—重庆:重庆大学出版社,2015.2

普通高等院校计算机基础教育系列教材

ISBN 978-7-5624-8835-4

I. ①C… II. ①刘…②熊… 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(2015)第 026878 号

高等院校规划教材
普通高等院校计算机基础教育系列教材

C 程序设计技术

主编 刘慧君 熊 壮

策划编辑 王 勇

责任编辑:文 鹏 版式设计:王 勇

责任校对:邹小梅 责任印制:赵 晟

重庆大学出版社出版发行

出版人:邓晓益

社址:重庆市沙坪坝区大学城西路 21 号

邮编:401331

电话:(023) 88617190 88617185(中小学)

传真:(023) 88617186 88617166

网址:<http://www.cqup.com.cn>

邮箱:fxk@cqup.com.cn(营销中心)

全国新华书店经销

重庆川外印务有限公司印刷

*

开本:787×1092 1/16 印张:20.5 字数:474 千

2015 年 2 月第 1 版 2015 年 2 月第 1 次印刷

印数:1—3 000

ISBN 978-7-5624-8835-4 定价:39.00 元

本书如有印刷、装订等质量问题,本社负责调换
版权所有,请勿擅自翻印和用本书
制作各类出版物及配套用书,违者必究

计算机程序设计能力是理工科大学生基本能力之一。培养学生的逻辑思维能力、抽象能力和基本的计算机程序设计能力,引导学生进入计算机程序设计的广阔空间,是大学计算机程序设计课程的主要目标。

计算机程序设计的主要任务,就是用合适的计算机程序设计语言,对解决问题的方法进行编码处理,即编制程序。C语言功能丰富、表达能力强、程序执行效率高、可移植性好,既有高级计算机程序设计语言的特点,同时又具有部分汇编语言的特点,故具有较强的系统处理能力。C语言是一种结构化程序设计语言,支持自顶向下、逐步求精的程序设计技术,通过C语言函数结构,可以方便地实现程序的模块化。在C语言基础之上发展起来的面向对象程序设计语言如C++、Java、C#等与C语言有许多共同特征。掌握C语言,对学习进而应用这些面向对象的程序设计语言有极大的帮助。

本书从结构化程序设计技术出发,以C程序设计语言为载体,通过对C语言的基本语法、语义以及学习C语言过程中各种常见典型问题的分析,通过对程序设计技术基础范畴内各种典型问题描述求解方法以及相应C语言代码的描述,展现了在程序设计过程中对问题进行分析、组织数据和描述解决问题的方法,展现了在计算机应用过程中将方法和编码相联系的具体程序设计过程,进而向读者介绍了计算机结构

化程序设计的基本概念、基本技术和方法。

本书的主要内容分为相辅相成的两个部分,第一部分包括第1章至第5章,主要介绍计算机程序设计高级语言共性的基础知识,包含的主要内容有基本数据类型的使用、运算符和表达式计算基础、标准库函数的使用方法和顺序程序设计、结构化程序设计、函数调用中的数值参数和地址值参数传递、变量的作用域和生存期、数组使用和字符串处理基础、函数调用中的数组参数传递、文件数据处理基础。对于需要了解和掌握结构化程序设计基本思想的读者,通过这部分的学习即可较为全面地掌握结构化程序设计的基本思想。第二部分包括第6章至第9章,主要介绍C语言特有的一些重要知识,包含的主要内容有返回指针值的函数、指向函数的指针以及指向函数指针变量作函数的形式参数、数组与指针的关系、指针数组、命令行参数、用指针实现动态数组的方法、结构体数据类型、联合体数据类型、用 typedef 关键字描述复杂数据类型、编译预处理基础、位运算与枚举类型。通过第二部分内容的学习,可以较为全面地掌握C语言的基础知识。

本书选用 Microsoft Visual C++ 6.0 作为教学环境,书中的所有教学示例都在 Microsoft Visual C++ 6.0 集成开发环境中通过。附录中还提供了 ASCII 码表、C 程序设计中常用的标准库函数、使用 Visual C++ 6.0 IDE 开发 C 程序的基本方法等重要学习资料。

本书适于作为高等院校各专业程序设计语言类课程教材,同时可作为计算机专业本专科学生、计算机应用开发人员、程序设计爱好者、计算机等级考试应试者在学习程序设计语言和程序设计技术时的参考教材。

参加本书编撰的作者都是长期从事程序设计课程教学的一线教师,在教材内容选取、章节教学顺序安排上都尽可能考虑了C语言基础

内容与初学程序设计难度上的平衡,通过丰富的例题、流畅的语言对教学内容进行了深入浅出的描述。参加本书编撰的教师有:陈策、何频、林宝如、刘慧君、冉春林、王欣如、伍星、熊壮、张全和,各章节编写分工如下:陈策(第4章),何频(第2章),林宝如(第5章),刘慧君、吴漪菡(第8章),冉春林(第9章),王欣如(第1章),伍星(第6章),熊壮(第7章),全书由刘慧君、熊壮进行内容调整、修改,统一定稿。

限于编者水平,书中不妥和错误之处在所难免,恳请读者不吝指教。

联系地址:重庆大学计算机学院。

E-Mail:lhjlc@cqu.edu.cn, xiongz@cqu.edu.cn

编者

2014年12月

目录

第1章 C程序设计初步	1
1.1 C程序结构和处理过程	1
1.1.1 C程序的基本结构	1
1.1.2 C程序的处理过程	3
1.2 C语言的基本数据类型	4
1.2.1 C程序中数据的表示	4
1.2.2 C语言基本数据类型	6
1.3 C语言基本运算符和表达式运算	18
1.3.1 C运算符和表达式概念	18
1.3.2 赋值运算符	20
1.3.3 算术运算符	20
1.3.4 自增自减运算符	21
1.3.5 复合赋值运算符	22
1.3.6 逗号运算符	23
1.3.7 sizeof运算符	24
1.3.8 数据类型转换	24
1.4 C语言标准库的使用方法	25
1.4.1 C标准库的使用方法	25
1.4.2 常用数学标准库函数介绍	26
习题1	30
第2章 C程序的基本控制结构	33
2.1 C语言关系运算和逻辑运算	33
2.1.1 关系运算符	33
2.1.2 逻辑运算符	33
2.2 分支结构程序设计	35
2.2.1 单分支程序设计	35
2.2.2 复合语句在程序中的使用	36
2.2.3 双分支程序设计	38
2.2.4 多分支程序设计	39
2.3 循环结构程序设计	44
2.3.1 while循环控制结构	44

2.3.2	do~while 循环控制结构	45
2.3.3	for 循环控制结构	47
2.3.4	空语句及其在程序中的使用	48
2.3.5	循环的嵌套结构	49
2.3.6	break 语句和 continue 语句	50
2.4	基本控制结构简单应用	53
2.4.1	穷举方法程序设计	53
2.4.2	迭代方法程序设计	55
2.4.3	一元高阶方程的迭代程序解法(*)	58
	习题 2	61
第 3 章	函数	67
3.1	函数的定义和调用	67
3.1.1	函数的定义和声明	67
3.1.2	函数调用中的数值参数传递	71
3.2	函数调用中的指针参数传递	74
3.2.1	指针变量的定义和引用	74
3.2.2	函数调用中的地址值参数传递	77
3.3	函数的嵌套调用和递归调用	80
3.3.1	函数的嵌套调用	80
3.3.2	函数的递归调用	82
3.4	变量的作用域和生存期	86
3.4.1	变量的作用域	87
3.4.2	变量的生存期	92
	习题 3	96
第 4 章	数组和字符串	103
4.1	数组的定义及数组元素的引用	103
4.1.1	一维数组的定义和元素引用方法	103
4.1.2	二维数组和 multidimensional 数组	106
4.2	字符数组和字符串	111
4.2.1	字符数组的定义和初始化	111
4.2.2	字符数组的输入输出	113
4.2.3	常用字符类数据处理标准库函数	117
4.3	函数调用中的数组参数传递	125
4.3.1	一维数组作函数的参数	125
4.3.2	二维数组作函数的参数	128
4.4	数组的简单应用	131
4.4.1	数组元素值的随机生成	131
4.4.2	基于数组的常用排序方法	132
4.4.3	基于数组的常用查找方法	135
	习题 4	139

第5章 C程序文件处理基础	145
5.1 文件处理的基本概念.....	145
5.1.1 C语言的文件数据类型.....	145
5.1.2 文件的打开/创建和关闭方法.....	147
5.1.3 文件内部读写位置指针和文件尾的检测方法.....	149
5.2 文件数据的读写方法.....	150
5.2.1 单个字符数据的读写.....	150
5.2.2 字符串数据的读写.....	154
5.2.3 格式化数据的读写.....	156
5.2.4 数据块的读写.....	159
5.3 随机存取文件处理基础.....	162
5.3.1 随机存取文件处理基本概念.....	162
5.3.2 随机存取文件数据的实现方法.....	162
习题5.....	170
第6章 指针	177
6.1 指针与函数.....	177
6.1.1 返回指针值的函数.....	177
6.1.2 指向函数的指针变量.....	179
6.2 指针与一维数组.....	185
6.2.1 指向一维数组元素的指针变量.....	185
6.2.2 指向一维数组的指针变量.....	186
6.3 指针与二维数组(*).....	190
6.3.1 多级指针的定义和引用.....	190
6.3.2 指向二维数组元素的指针变量.....	191
6.3.3 指向二维数组的指针变量.....	192
6.4 指针数组与命令行参数.....	196
6.4.1 指针数组的定义和引用.....	197
6.4.2 命令行参数.....	199
6.5 使用指针构建动态数组.....	201
6.5.1 动态数据的概念和存储分配标准库函数.....	201
6.5.2 一维动态数组的建立和使用.....	203
6.5.3 二维动态数组的建立和使用.....	205
6.6 指针与字符串.....	208
6.6.1 字符串的指针表示.....	208
6.6.2 字符串处理标准函数的指针参数.....	210
习题6.....	219
第7章 结构体和联合体数据类型	225
7.1 结构体类型的定义和使用.....	225
7.1.1 结构体类型和结构体变量的定义.....	225

7.1.2	typedef 关键字的简单应用	228
7.1.3	结构体变量的使用方法	230
7.2	结构体数组	234
7.2.1	结构体数组的定义和数组元素引用	234
7.2.2	结构体数组作函数的参数	236
7.3	结构体数据类型与指针的关系	238
7.3.1	结构体类型变量与指针的关系	238
7.3.2	结构体类型数组与指针的关系	240
7.3.3	结构体类型简单应用——单链表基本操作(*)	242
7.4	联合体数据类型	249
7.4.1	联合体数据类型的定义及联合体变量的引用	249
7.4.2	联合体类型与结构体类型的区别	253
习题 7		256
第 8 章	编译预处理基础	263
8.1	宏定义预处理命令及其简单应用	263
8.1.1	不带参数的宏定义	263
8.1.2	带参数的宏定义	266
8.2	文件包含预处理命令及其简单应用	269
8.2.1	文件包含书写形式及意义	269
8.2.2	用文件包含方式组织多源文件 C 程序	269
8.3	条件编译预处理命令及其简单应用	272
8.3.1	#if、#elif、#else、#endif	272
8.3.2	#ifdef 和 #ifndef	273
习题 8		276
第 9 章	枚举类型和位运算	283
9.1	枚举类型及其简单应用	283
9.1.1	枚举类型的定义和枚举变量的引用	283
9.1.2	枚举数据类型的简单应用	285
9.2	位运算及其应用	288
9.2.1	位运算符	288
9.2.2	位运算的简单应用	292
习题 9		296
附录 A	ASCII 码表(基本表部分 000~127)	301
附录 B	C 语言中的保留字	303
附录 C		303
附录 D		310
参考文献		317

C 程序设计初步

1.1 C 程序结构和处理过程

C 语言是 1978 年由美国电话电报公司(AT&T)贝尔实验室正式发表,早期主要用于 UNIX 系统。由于 C 语言具有强大的功能,很快移植到其他操作系统平台下,在各类大、中、小和微型计算机上得到了广泛使用,成为当代最优秀的程序设计语言之一。C 语言有如下的一些主要特点:

①简洁、紧凑,使用方便、灵活。C 语言只有 30 多个关键字,9 种控制语句,程序书写形式自由。

②运算类型丰富。C 语言共有 34 种运算符,其中包括括号、赋值、逗号等,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数据类型丰富。C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型以及枚举类型等,可以表达各种复杂的数据结构,具有很强的数据处理能力。

④完全模块化和结构化的语言。C 语言具有结构化的控制语句,用函数作为程序的单元,便于实现程序的模块化。

⑤语法规则简洁。C 语法限制不太严格,程序设计和书写形式自由度大。

⑥C 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。

⑦生成目标代码质量高,程序执行效率高。

⑧可移植性好,程序容易在新的系统获得使用。

1.1.1 C 程序的基本结构

下面通过两个完整的 C 程序了解 C 语言源程序结构的特点。在这两个示例中,主要了解组成一个 C 源程序的基本部分和书写格式,程序实现功能的细节,在后面的章节中逐步介绍。

[例 1.1] 在屏幕上输出信息:“This is my first C program.”。

```
/* Name: ex0101.cpp */
#include <stdio.h> //这是编译预处理命令
int main()
{
    printf("This is my first C program.\n");
    return 0;
}
```

[程序代码说明]

①程序仅由一个主函数组成,main 是主函数的函数名,函数名前面的 int 表示函数是整型类型函数。这种类型的主函数必须在函数的最后使用 C 语句“return 0;”来表示程序正常结束。C 程序最常用的两种主函数框架如下所示,本书采用整型主函数形式。

```
#include <stdio.h>                                #include <stdio.h>
int main()                                         void main()
{
    ...                                           {
    return 0;                                       ...
}
```

②程序中的注释。注释部分仅起到说明作用,程序不执行注释部分。在现代的开发环境下,注释的书写有多行注释方式和单行注释方式两种。用符号/*和*/作为括号的是多行注释方式,其中的注释内容既可以写在一行上,也可以写在多行上;用符号//作为引导的是单行注释方式,其注释内容只能书写在一行上。

③程序中调用输出函数 printf,将括号中的字符串内容显示到屏幕(显示器)上,这也是 C 程序输出程序运行结果的主要方式。为了能够使用标准函数 printf,在程序前面使用了预处理语句:#include <stdio.h>。

[例 1.2] 从键盘上输入两个整数,求它们中间的较大者。

```
/* Name: ex0102.cpp */
#include <stdio.h>
int main() //主函数
{
    int max(int a,int b); //函数声明*/
    int x,y,z; //变量说明*/
    printf("请用户输入两个整数,用空格分隔:\n");
    scanf("%d%d",&x,&y); //输入 x,y 值,注意书写格式*/
    z=max(x,y); //调用 max 函数*/
    printf("max num: %d\n",z); //输出程序执行结果*/
    return 0;
}
int max(int a,int b) //定义 max 函数
```

```

}
int c;
if(a > b)
    c = a;
else
    c = b;
return c;    // 把结果返回主调函数
}

```

[程序代码说明]

①程序执行时,由用户输入两个整数(用空格分隔,按回车键结束输入),程序执行后输出其中较大的数。

②程序由主函数 main 和函数 max 组成。max 函数是一个用户自定义函数,其功能是比较从主函数传递过来的两个数,然后把较大的数返回给主函数。

③程序中,函数之间是并列关系。C 程序中,每个函数都是一个相对独立的代码块,它们在程序中书写的顺序是任意的。被调用的函数定义如果出现在调用点之后,必须要对被调函数进行声明。例如在本例中,主函数 main 调用 max 函数,max 函数的定义在主函数之后,所以在主函数中必须对 max 函数进行声明(参见主函数中的函数声明语句)。关于函数有关知识,将在第 3 章进行详细介绍。

通过以上例子可以看出,C 程序在结构上有如下特点:

①一个 C 程序可以由一个或多个函数组成,但无论有多少个函数,都有且仅有一个主函数(即 main 函数)。

②函数由函数头和函数体两个部分构成,用花括号 {} 包括函数体部分。函数体由一系列 C 语句组成,用以实现函数的功能。

③C 程序根据需要可以有预处理语句,预处理语句通常放在源程序的最前面。由于每个完整的 C 程序都会涉及数据的输入、输出,所以每一个 C 程序都一定含有预处理语句:#include <stdio.h>或者#include "stdio.h"。

④C 程序中,每一条语句都必须以分号结尾。但对于预处理命令,函数头和花括号“{}”之后不需要加分号。

⑤标识符、保留字以及各语言独立成分之间至少用一个空格以示间隔。

⑥程序应当包含适当的注释,以增强程序的易读性和可理解性。

1.1.2 C 程序的处理过程

计算机不能直接识别和执行用高级语言书写的指令,需要通过“解释方式”或者“编译方式”对程序进行处理。在用编译方式处理高级语言编写的程序时,需要经过“编译”和“连接”两个过程。

用符合 C 语言规范的方式书写并保存的 C 程序称为源程序文件。处理 C 源程序文件时,首先使用“编译程序”把 C 源程序翻译成二进制形式的目标码,并用目标文件的形式存放;再通过“连接程序”将源程序中用到的其他功能函数与目标程序连接在一起,最后生成

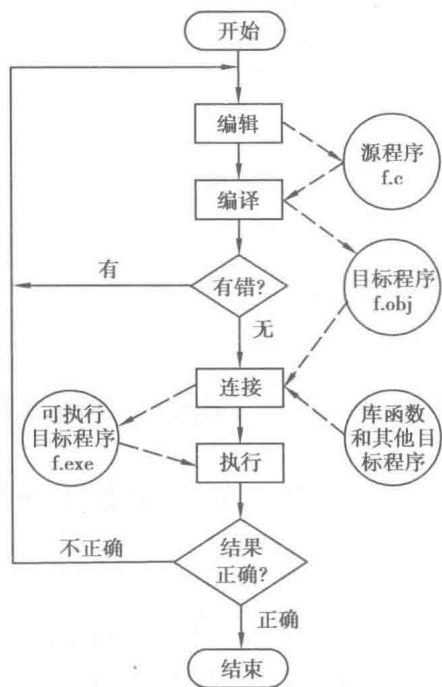


图 1.1 程序的处理过程

计算机系统可以辨识和执行的机器语言程序。

以 Microsoft Visual C++ 6.0 集成环境(简称 VC++6.0 IDE)为例(如图 1.1 所示),一个 C 程序的处理过程为:

①编写源程序代码。在开发环境中,使用其编辑功能输入和修改源程序,获得程序的源程序文件(后缀为.c 或.cpp)。

②编译程序。通过执行开发环境中的“编译”命令,对源程序进行编译处理。编译处理的主要工作是检查程序中是否存在语法错误。若有语法错误,则返回编辑环境修改程序;若没有语法错误,则生成由二进制代码表示的目标程序文件(后缀为.obj)。

③连接程序。获得程序的目标文件后,通过开发环境中的“连接”命令,将目标文件和由开发环境提供的标准库函数、启动代码等进行连接,形成可执行文件(后缀为.exe)。

④运行与调试。获得执行文件后,在开发环境中,通过使用“运行”命令即可执行程序。若程序执行后获取了正确结果,则程序处理过程完成;若程序执行后没有获得期望的结果,表示该程序有逻辑错误(即程序代码不能实现要求的功能),这时需要调试程序,即返回编辑环境,寻找错误出现的原因,修改程序代码。

1.2 C 语言的基本数据类型

C 语言的数据包含两个大类:基本数据类型和构造数据类型。基本数据类型也称为内置数据类型,该类数据可以在程序中直接使用。

1.2.1 C 程序中数据的表示

C 程序的基本构成成分包括字符集,标识符,保留字,常量,变量,运算符等。

1) C 语言的字符集

每种计算机程序设计语言都规定了在书写源程序时允许使用的字符集,以便语言处理系统能正确识别它们。C 语言规定书写 C 源程序的字符集由以下字符组成:

- 小写英文字母: a b c … z
- 大写英文字母: A B C … Z
- 数字字符: 0 1 2 3 … 9
- 特殊字符: + = - _ () * & % \$! | < > . , ; : “ ‘ / ? { } ~ [] ^
- 不可印出字符: 空格、换行、制表符等

2) 标识符和保留字

标识符是在程序中为数据对象命名的单词(字符序列),分为两大类:系统保留字和用户标识符。

C 语言有 37 个系统保留字(又称为关键字)。保留字是一类特殊的标识符,在 C 语言中具有特定严格意义的基本词汇,任何情况下都不能将它们作为用户标识符使用。下面列出的是 C 语言中的保留字(其中标有“*”的是在 C99 标准中增加的):

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

还有几个标识符从严格意义上说不属于系统保留字,它们常出现在 C 的预处理器中,C 语言开发环境中为它们赋予了特定的含义,建议用户不要将它们在程序中随意使用,以免造成混淆。这些标识符是:

define	undef	include	ifdef
ifndef	endif	line	error
elif	pragma		

用户标识符是在程序中给用户自定义数据对象(如变量、常量、函数、数据类型等)命名的符号。在不混淆的情况下,用户标识符简称为标识符。标识符的命名规则是:

- ①构成标识符的字符只能是字母、数字和下划线。
- ②标识符中第一个符号不能是数字,只能是字母或者下划线。
- ③标识符构成时要区分字母的大小写,即 abc 和 ABC 是不相同的标识符。
- ④不能用保留字作用户标识符。

在程序中自定义标识符时,除了必须遵守标识符的命名规则外,还需要注意以下两个方面:一是要将标识符取得既有意义,又便于阅读;二是要注意避免含义上或书写时引起混淆。

下面列出的是一些合法的用户自定义标识符例子:

```
a x1 file_name _buf PI
```

下面列出的是不合法的用户自定义标识符例子及错误原因:

```
123abc /* 不是以英文字母开头 */
```

```
float /* 与系统保留字同名 */
```

```
up-to          /* 标识符中出现了非法字符“-” */
zhang san     /* 标识符中间出现了非法字符空格 */
```

3) 常量和变量

常量和变量是计算机高级语言程序中数据的两种表现形式。在程序执行过程中,数据的值不能发生改变,则称其为常量;数据的值有可能发生改变,则称其为变量。

在程序中,常量可以不经说明而直接引用(即直接书写在程序代码中),而变量则必须先定义后使用。变量在程序中用标识符命名,同时变量还对应计算机系统存储器中的某一段存储空间(容纳该变量值的空间)。程序中定义变量时,不但要用合适的标识符命名,而且还必须说明变量的数据类型,以便编译器确定该变量的存储和处理方法。变量定义的一般形式为:

数据类型名 变量名 1,变量名 2,……;

1.2.2 C 语言基本数据类型

C 语言中,数据类型可分为 4 类,即基本数据类型、构造数据类型、指针类型、空类型,如图 1.2 所示。

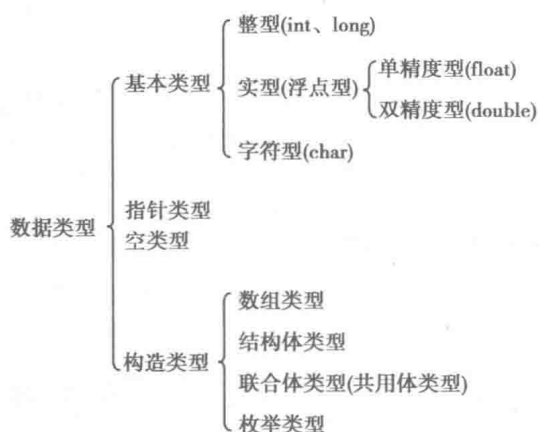


图 1.2 C 语言的数据类型

1) 整型数据

整型数据是计算机程序设计中最常用的数据类型之一。在 C 语言中,整型数据用机器的一个字长来存储,所以整型数据的表示范围与计算机系统的软硬件环境有关。在字长为 16 位计算机系统中,整型数据占用 2 个字节,表示的数据范围为 $-32\ 768 \sim 32\ 767$ ($-2^{15} \sim 2^{15} - 1$)。在字长为 32 位的计算机系统中,整型数据占用 4 个字节,表示的数据范围为 $-2\ 147\ 483\ 648 \sim 2\ 147\ 483\ 647$ ($-2^{31} \sim 2^{31} - 1$)。

整型常量在 C 程序中有 3 种书写形式:

- ①十进制整数:如 12、-123、0 等。
- ②八进制整数:以 0 开头的整数是八进制整型常量,如 0777、-011 等。
- ③十六进制整数:以 0x 开头的整数是十六进制整型常量,如 0x123、0xff 等。

在 C 程序中,用于存放整型数据的变量称为整型变量。整型变量有基本整型、短整型、长整型和无符号整型四种,其数据类型名分别由关键字 int、short、long 与 unsigned 表示。下面是一些 C 程序中整型变量定义的示例:

```
int a,b,c;    //定义 a,b,c 为整型变量
long x,y;     //定义 x,y 为长整型变量
unsigned p,q; //定义 p,q 为无符号整型变量
```

变量定义时可以进行初始化,即在定义的同时通过赋值号(=)将值赋值给变量,例如:语句 int a=100,b=200;等。

一个程序中,数据的输入或输出,特别是输出是必不可少的。C 程序中的数据输入、输出一般通过标准库函数 `scanf` 和 `printf` 来实现。

C 程序中,可以通过调用标准函数 `scanf` 实现整型数据的输入。C 语言规定必须使用变量的取地址形式,一般使用控制符号 `%d`、`%hd`、`%ld`、`%u` 进行输入控制。下面代码段描述了整型数据输入 C 语句的书写示例:

```
int abc1;
short abc2;
long abc3;
unsigned abc4;
scanf("%d",&abc1);           //为整型变量 abc1 输入数据,注意变量的取地址形式
scanf("%hd",&abc2);         //为短整型变量 abc2 输入数据
scanf("%ld",&abc3);         //为长整型变量 abc3 输入数据
scanf("%u",&abc4);          //为无符号整型变量 abc4 输入数据
scanf("%d%u",&abc1,&abc4);  //为变量 abc1 和 abc4 输入数据,数据间用空格分隔
scanf("%d,%u",&abc1,&abc4); //为变量 abc1 和 abc4 输入数据,数据间用逗号分隔
```

C 程序中,可以通过调用标准函数 `printf` 实现整型数据输出,一般使用控制符号 `%d`、`%hd`、`%ld`、`%u` 进行输出控制。下面代码段描述了整型数据输出 C 语句的书写示例:

```
printf("%d\n",abc1);        //输出变量 abc1 的值,然后换行
printf("%ld",abc3);        //输出变量 abc3 的值(不换行),注意比较上一个输出控制
printf("%d\n%u\n",abc1,abc4); //输出变量 abc1 和变量 abc4 的值,数据分两行输出
printf("%d,%u\n",abc1,abc4); //输出变量 abc1 和变量 abc4 的值,输出数据间用逗号分隔
printf(" abc2=%hd\n",abc2); //先输出字符序列:abc2=,然后接着输出变量 abc2 的值
```

[例 1.3] 整型数据在程序中的使用示例。

```
// Name:ex0103.cpp
#include <stdio.h>
int main()
{
    int a,b,c,d;
    a=1000;    //常量值直接赋给变量
    printf("请输入变量 b 和 c 的值:");    //程序运行时提示用户输入数据
    scanf("%d,%d",&b,&c);    //输入两个数据,用逗号分隔
    d=a+b+c+1000;    //常量值直接写在表达式中
```