



高等院校计算机教材系列

C/C++ 程序设计实验教程

秦维佳 侯春光 孟艳红 伞宏力 编著



机械工业出版社
China Machine Press

C/C++

程序设计实验教程

张世英 编 王 强 主审 中国铁道出版社

中国铁道出版社

高等院校计算机教材系列

TP312

2230C

2007

C/C++ 程序设计实验教程

秦维佳 侯春光 孟艳红 伞宏力 编著



机械工业出版社
China Machine Press

本书是《C/C++ 程序设计教程》的配套学习辅导用书。

全书共分 12 章, 内容包括 C/C++ 语言程序设计初步, C 语言的数据类型及表达式, 结构化程序设计, 数组、函数与指针的概念和实例, 结构体与共用体, 位运算与文件的概念, 面向对象程序设计, 类的继承与多态, C++ 的 I/O 流。每章包括知识体系、学习要点、上机实训、习题和参考答案。

本书可供初学者学习 C 语言, 也可供参加计算机等级考试的应试者考前复习和参考。

版权所有, 侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

C/C++ 程序设计实验教程/秦维佳等编著. —北京: 机械工业出版社, 2007. 4
(高等院校计算机教材系列)

ISBN 978-7-111-20610-1

I. C… II. 秦… III. C 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字(2006)第 159358 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

策划编辑: 武恩玉

责任编辑: 杨庆燕

北京慧美印刷有限公司印刷 · 新华书店北京发行所发行

2007 年 2 月第 1 版第 1 次印刷

184mm × 260mm · 11.25 印张

定价: 18.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

本社购书热线: (010)68326294

前 言

计算机应用能力是新世纪人才不可缺少的基本素质。程序设计是工科各专业计算机应用能力培养的重要技术基础，而 C/C++ 是目前国内外广泛使用的一种程序设计语言，也是国内外大学讲授程序设计方法的首选语言。

C/C++ 是一门实践性很强的课程，在努力提高课堂教学的同时，必须大力加强实践教学环节的要求和管理，加强上机实验和练习题的训练。为了方便教学，学练结合，学以致用，我们编写了《C/C++ 程序设计实验教程》一书，作为《C/C++ 程序设计教程》的配套学习辅导用书。

本书内容丰富、注重实践；由浅入深、便于理解；案例广泛、通俗易懂、便于自学。全书共分 12 章。第 1 章介绍 C/C++ 语言程序设计初步；第 2 章介绍 C 语言的数据类型及表达式；第 3 章介绍结构化程序设计方法；第 4 章介绍数组的概念和实例；第 5 章介绍函数的概念和实例；第 6 章介绍指针的概念和实例；第 7 章介绍结构体与共用体；第 8 章主要介绍位运算；第 9 章介绍文件的概念；第 10 章介绍面向对象程序设计；第 11 章介绍类的继承与多态；第 12 章介绍 C++ 的 I/O 流。本书给出了每章的知识结构、学习要点、上机实训、习题及参考答案，供读者学习时借鉴和参考。

本书由秦维佳组织编写。其中第 1~3 章由秦维佳编写；第 4~5 章由孟艳红编写；第 6~9 章由侯春光编写；第 10~12 章由伞宏力编写。秦维佳对全书进行了统编、程序调试与定稿。在教材编写过程中寿业勇、于彤彤、赵翠红、赵爽等审读了初稿，并提出了许多宝贵意见，在此表示衷心感谢。

由于编者水平有限，错误在所难免，请广大读者批评指正。

作 者

2006 年 10 月

目 录

前言

第1章 C语言及程序设计初步 1

- 1.1 知识体系 1
- 1.2 学习要点 1
- 1.3 上机实训 2
- 1.4 习题 4
- 1.5 参考答案 4

第2章 数据类型、运算符和表达式 5

- 2.1 知识体系 5
- 2.2 学习要点 6
 - 2.2.1 基本符号和标识符 6
 - 2.2.2 数据类型 6
 - 2.2.3 常量与变量 7
 - 2.2.4 数据的输入输出 7
 - 2.2.5 运算符 8
 - 2.2.6 表达式 9
- 2.3 上机实训 9
- 2.4 习题 11
- 2.5 参考答案 14

第3章 结构化程序设计 16

- 3.1 知识体系 16
- 3.2 学习要点 17
 - 3.2.1 C语言基本语句 17
 - 3.2.2 结构化程序设计 17
- 3.3 上机实训 19
 - 3.3.1 分支程序实验 19
 - 3.3.2 循环程序实验 20
 - 3.3.3 综合实验 20
- 3.4 习题 22
- 3.5 参考答案 26

第4章 数组 28

- 4.1 知识体系 28
- 4.2 学习要点 28

- 4.2.1 一维数组 28

- 4.2.2 二维数组和 multidimensional array 29

- 4.2.3 字符数组和字符串 30

4.3 上机实训 32

- 4.3.1 一维数组实验 32

- 4.3.2 二维数组实验 33

- 4.3.3 字符数组实验 35

- 4.3.4 综合实验 36

4.4 习题 40

4.5 参考答案 46

第5章 函数 54

- 5.1 知识体系 54

- 5.2 学习要点 54

- 5.2.1 函数的定义 54

- 5.2.2 函数的调用 55

- 5.2.3 函数与数组 56

- 5.2.4 局部变量和全局变量 56

- 5.2.5 变量的存储类别 57

- 5.2.6 内部函数和外部函数 58

5.3 上机实训 58

- 5.3.1 函数调用方法实验 58

- 5.3.2 数组作函数参数实验 60

- 5.3.3 嵌套调用函数和递归调用
函数实验 62

- 5.3.4 变量存储类别实验 64

- 5.3.5 综合实验 66

5.4 习题 68

5.5 参考答案 74

第6章 指针 81

- 6.1 知识体系 81

- 6.2 学习要点 82

- 6.2.1 指针的概念 82

- 6.2.2 指针与数组 83

- 6.2.3 指针与字符串 84

6.2.4 指针与函数	84	7.3.6 链表操作	109
6.2.5 指向指针的指针	86	7.3.7 共用体	111
6.3 上机实训	86	7.3.8 枚举类型	112
6.3.1 指针的运算	86	7.3.9 用 typedef 定义类型	113
6.3.2 指针与一维数组	87	7.4 习题	113
6.3.3 指针与二维数组	88	7.5 参考答案	115
6.3.4 指针数组	89	第 8 章 位运算	117
6.3.5 指针与字符串	89	8.1 知识体系	117
6.3.6 指针作为函数参数	90	8.2 学习要点	117
6.3.7 数组指针作为函数参数	91	8.2.1 位运算符的使用方法	117
6.3.8 字符串指针作为函数参数	91	8.2.2 位段	118
6.3.9 指向函数的指针	92	8.3 上机实训	118
6.3.10 返回指针值的函数	93	8.3.1 按位“与”运算符(&)	118
6.3.11 main() 函数中的参数	94	8.3.2 按位“或”运算符()	118
6.3.12 指向指针的指针	94	8.3.3 按位“取反”运算符(~)	119
6.4 习题	95	8.3.4 按位“异或”运算符(^)	119
6.5 参考答案	98	8.3.5 “左移”运算符(<<)	119
第 7 章 结构体与共用体	101	8.3.6 “右移”运算符(>>)	120
7.1 知识体系	101	8.3.7 位段	120
7.2 学习要点	101	8.4 习题	121
7.2.1 结构体类型的定义	101	8.5 参考答案	122
7.2.2 结构体变量的定义与 初始化	102	第 9 章 文件	123
7.2.3 结构体变量的引用	103	9.1 知识体系	123
7.2.4 结构体数组	103	9.2 学习要点	123
7.2.5 结构体指针	104	9.2.1 文件概述	123
7.2.6 链表	104	9.2.2 文件的打开与关闭	124
7.2.7 共用体	105	9.2.3 文件的顺序读写	124
7.2.8 枚举类型	105	9.2.4 文件的定位与随机读写	125
7.2.9 用 typedef 定义类型	106	9.2.5 文件操作的出错检测	126
7.3 上机实训	106	9.3 上机实训	126
7.3.1 结构体变量的定义与 初始化	106	9.3.1 文件的打开与关闭	126
7.3.2 结构体变量在函数间的 传递	106	9.3.2 输入和输出一个字符	126
7.3.3 结构体数组的应用	107	9.3.3 输入和输出一个字符串	127
7.3.4 结构体指针的应用	108	9.3.4 格式化的输入和输出	127
7.3.5 结构体指针作函数参数	108	9.3.5 按数据块的形式输入和 输出	128
		9.3.6 文件的定位与随机读写	129
		9.4 习题	129

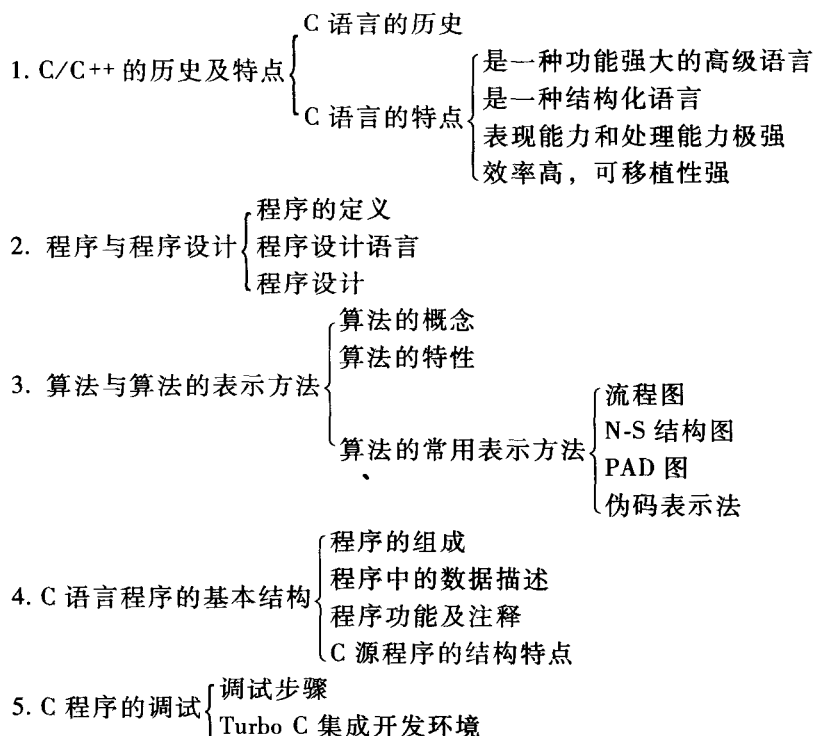
9.5 参考答案	131	11.3.1 类的派生与继承实验	147
第 10 章 面向对象程序设计及 C++ 基础	132	11.3.2 多态性实验	149
10.1 知识体系	132	11.3.3 综合实验	152
10.2 学习要点	133	11.4 习题	153
10.2.1 C++ 对 C 的扩充	133	11.5 参考答案	157
10.2.2 C++ 中的类与对象	134	第 12 章 C++ 的输入输出流	161
10.2.3 构造函数与析构函数	135	12.1 知识体系	161
10.3 上机实训	136	12.2 学习要点	161
10.3.1 C++ 基础实验	136	12.2.1 输入输出的格式控制	161
10.3.2 C++ 的类与对象实验	137	12.2.2 用户自定义类型的输入 输出	163
10.3.3 综合实验	139	12.2.3 文件的输入输出	163
10.4 习题	140	12.3 上机实训	164
10.5 参考答案	142	12.3.1 C++ 的基本输入输出 实验	164
第 11 章 类的继承与多态	145	12.3.2 C++ 的文件 IO 实验	165
11.1 知识体系	145	12.3.3 综合实验	167
11.2 学习要点	146	12.4 习题	168
11.2.1 继承与派生	146	12.5 参考答案	170
11.2.2 多态性	147	参考文献	173
11.3 上机实训	147		

第1章 C语言及程序设计初步

C语言是目前世界上流行、使用最广泛的高级程序设计语言。C语言的主要特色是兼顾了高级语言和汇编语言的特点，简洁、丰富、可移植。目前有许多应用程序都是用C语言来编写的，例如，数据结构、数据库系统、人机接口控制、数值分析、图像处理等，C语言受到人们的极大推崇。

本章将带你走进C语言的世界，初步了解C语言的功能，教你编写第一个C语言程序。在这一过程中，读者要了解C/C++的发展简史，理解算法的基本概念，了解结构化程序设计方法，了解C语言程序的基本结构，掌握C语言的程序调试步骤。

1.1 知识体系



1.2 学习要点

1. C/C++ 的历史及特点

初步了解C语言的历史及特点。

2. 程序与程序设计

程序设计是一门技术，需要相应的理论、技术、方法和工具来支持。

程序的定义：在计算机中，程序是计算机的一组指令。程序是计算机执行后，能完成特定功能的指令序列。

程序设计语言：是程序设计的工具。

程序设计：根据问题的需求，设计数据结构和算法，编制程序和调试程序，使计算机程序能完成所需要的任务。程序首先应能正确完成任务，并且是可靠的。简单地说，程序设计

是设计和编制程序的过程。

3. 算法与算法的表示方法

程序设计离不开算法，程序 = 算法 + 数据结构。算法是指解决问题的方法和步骤。

算法的概念：为解决一个实际问题而采取的方法和步骤称为“算法”。

算法的特性：

- 1) 可行性。
- 2) 确定性。
- 3) 有穷性。
- 4) 输入。
- 5) 输出。

算法的常用表示方法：

1) 流程图：流程图是用来表示算法的图形，是软件制作过程的设计表示工具，流程图表达直观、清晰，易于学习和掌握。

2) N-S 图：N-S 图是用方框图来代替传统的程序流程图的技术。在这种流程图中把流程线全部去掉，全部算法写在一个矩形框内，在框内还可以包含其他框。

3) PAD 图：是问题分析图 (Problem Analysis Diagram)，是继程序流程图和方框图之后，提出的又一种描述算法的图形工具。

4) 伪代码：是介于自然语言和计算机语言之间的文字和符号，用来表示算法，即与程序设计语言很相近的伪码来描述处理过程的细节，如结构化语言。

4. C 语言程序的基本结构

- 1) 程序的组成。
- 2) 程序中的数据描述，程序中使用的变量要预先定义。
- 3) 适当的、正确的注释可以帮助读者正确地理解程序。

4) C 源程序的结构特点：

- 一个 C 语言源程序可以由一个或多个源程序文件组成。
- 每个源程序文件可由一个或多个函数组成。
- 一个源程序不论由多少个文件组成，都有一个且只能有一个 main 函数，即主函数。
- 源程序中可以有预处理命令 (include 命令)，预处理命令通常应放在源文件或源程序的最前面。
- 每一个注释都必须用斜杠加星号括起来 (/ * */)，每一个语句都必须以分号结尾。但预处理命令、函数头和花括号 “}” 之后不能加分号。
- 标识符、关键字之间必须至少加一个空格以示间隔。

5. C 程序的调试步骤

第一步：编辑源程序。

第二步：编译源程序。

第三步：连接。

第四步：运行。

1.3 上机实训

1. 实验目的

- 1) 掌握 Turbo C 3.0 集成环境的进入与退出。
- 2) 掌握 Turbo C 3.0 集成环境各种窗口的切换。
- 3) 了解 Turbo C 3.0 集成环境的设置。

- 4) 了解 Turbo C 3.0 与 Turbo C 2.0 集成环境的操作区别。
- 5) 掌握 C 语言源程序的建立、编辑、修改、保存、编译、连接和运行。

2. 实验内容

- 1) 开机，进入 DOS 操作平台或 Windows 平台。
- 2) 进入 Turbo C 3.0 集成环境。
- 3) 了解 Turbo C 3.0 集成环境及其菜单的使用。
- 4) 了解并掌握 Turbo C 3.0 集成环境的设置。
- 5) 输入 C 语言源程序，并以 lab1.c 为文件名存盘。

源程序清单如下：

```
#include <stdio.h>
void main()
{
    printf("welcome!\n");
}
```

- 6) 在 Turbo C 3.0 中打开 lab1.c，试编译、连接、运行该程序。人们已经习惯使用鼠标操作，而 Turbo C 3.0 恰好支持鼠标操作。
- 7) 选择主菜单 Windows 下的 Output 选项，查看运行结果，如图 1-1 所示。

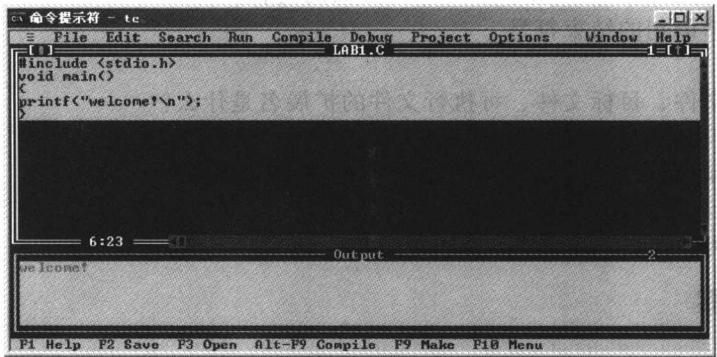


图 1-1 程序运行示例 1

- 8) 再新建一个源程序文件，将输入的两个数从小到大排序，并进行调试运行，注意编译时的错误信息。

源程序清单如下：

```
#include <stdio.h>
void main()
{
    int a,b,c;
    printf("input a=%d");
    scanf("%d",&a);
    printf("input b=%d");
    scanf("%d",&b);
    if(a>b){c=a;a=b;b=c;}
    printf("a=%d\tb=%d\n",a,b);
}
```

程序运行结果如图 1-2 所示。

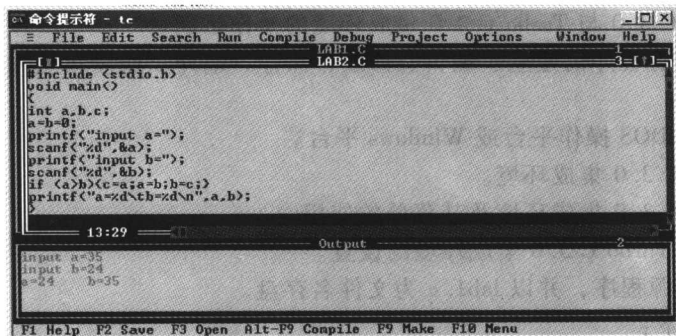


图 1-2 程序运行示例 2

1.4 习题

一、填空题

1. 每个 C 语言编写的程序必须有一个，并且只能有一个_____函数。
2. 一个 C 语言程序总是从_____位置开始执行。
3. C 语言的基本单位是_____。
4. 一个程序应当包含_____和_____两方面的内容。
5. C 语言中，每个语句的结束符是_____。

二、问答题

1. C 语言源程序文件、目标文件、可执行文件的扩展名是什么？
2. 试叙述 C 语言程序的基本构成。
3. 何谓算法？何谓程序？何谓程序设计语言？何谓程序设计？
4. 算法的常用表示方法有哪些？
5. C 语言以函数为程序的基本单位，有什么优点？

1.5 参考答案

一、填空题

1. 主
2. main()
3. 函数
4. 算法 数据结构
5. 分号或；

二、问答题

1. C 语言源程序文件的扩展名为 .c，目标文件的扩展名为 .obj，可执行文件的扩展名为 .exe。
2. C 语言程序由函数组成，可以包含多个函数，但一个 C 语言程序有且只有一个主函数。
3. 算法：为解决一个实际问题而采取的方法和步骤称为“算法”；
程序：程序是供计算机执行后，能完成特定功能的指令序列；
程序设计语言：是一种编程用的计算机语言，是程序设计的工具。
程序设计：根据问题的需求，设计数据结构和算法，编制程序和调试程序，使计算机程序能完成所需要的任务。简单地说，程序设计是设计和编制程序的过程。
4. 流程图、N-S 图、PAD 图和伪代码等。
5. 便于实现结构化程序设计。

* 表示该习题在《C/C++ 程序设计教程》一书中也存在。

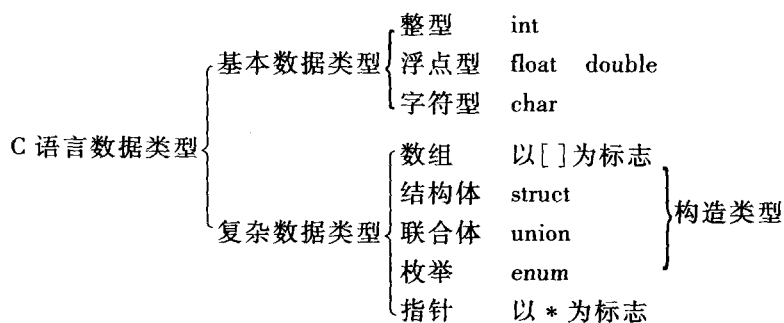
第2章 数据类型、运算符和表达式

通过本章的学习要掌握基本数据的各种表示方法；掌握算术运算、关系运算、逻辑运算、赋值运算等运算的意义；掌握表达式的书写方法；掌握整数、浮点数、字符、字符串的格式输入和输出方法。为程序设计打下一个良好的基础。

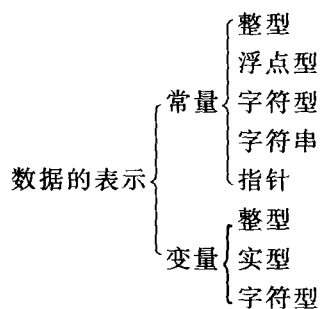
2.1 知识体系

1. 基本符号和标识符

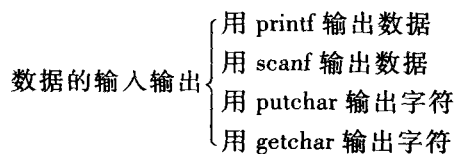
2. 数据类型



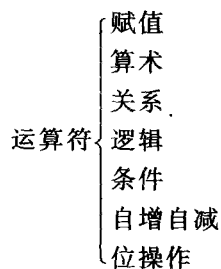
3. 常量与变量



4. 数据的输入输出



5. 运算符



6. 表达式

表达式 { 算术
关系
逻辑
赋值
条件
逗号

2.2 学习要点

2.2.1 基本符号和标识符

1. C 语言的基本符号由以下 7 类

- 数字：0~9 十个数字。
- 英文字母：大写字母 26 个(A~Z)；小写字母 26 个(a~z)。
- 括号：(), [], {} 三种。
- 运算符：+ - * / = > < % & ! ? ! ^ ~ 14 种。
- 标点符号：' " ; : 4 种。
- 特殊符号：\ _ \$ # 4 种。
- 空白符：空格符、换行符、制表符三种。

2. C 语言的标识符分三类：关键字、特定字、用户定义字

1) 关键字(也称保留字) 关键字是一些英文单词，一般用小写字母表示。C 语言利用单词的意义标识 C 程序结构。关键字是 C 编译程序预先登录的标识符，它们代表固定的含义，用户不能随便用来标识程序对象。下面是 C 语言的大部分关键字：

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

2) 特定字 特定字是具有特殊含义的标识符，它们虽然不是 C 语言的关键字，但是习惯上把它们看成关键字。下面几个特定字通常在 C 程序中的预处理命令行中使用。

define undef include ifdef ifndef endif line elif

3) 用户定义字 用户定义字是程序员在程序设计时为变量、常量、函数等程序对象起的名字。

2.2.2 数据类型

基本数据类型见表 2-1。

表 2-1 基本数据类型的存储方式和取值范围

数据类型	类型符	字节	数值范围
整型	int	4	-2 147 483 648 ~ 2 147 483 647
长整型	long int	4	-2 147 483 648 ~ 2 147 483 647
短整型	short int	2	-32 768 ~ 32 767
无符号整型	unsigned int	4	0 ~ 4 294 967 295
无符号短整型	unsigned short int	2	0 ~ 65 535
单精度浮点型	float	4	3.4E-38 ~ 3.4E38
双精度浮点型	double	8	1.7E-308 ~ 1.7E308
长双精度浮点型	long double	8	1.7E-308 ~ 1.7E308
字符型	char	1	0 ~ 255(256 个字符)

2.2.3 常量与变量

1. 整型常量

整型常量的表示通常有三种：十进制表示、八进制表示、十六进制表示。十进制表示不能以 0 开头；八进制表示以数字 0 开头；十六进制以数字 0 和 x 开头。

2. 浮点型常量

在数学上，表示小数由两种方式：一种是小数表示法，另一种是科学计数法，也称指数表示法。浮点数表示法来源于数学中的指数表示形式，如 193.5 可以表示为 0.1935×10^3 或 1.935×10^2 等。

3. 字符型常量

在 C 语言中的字符常量是用单引号括起来的一个字符。

4. 字符串常量

在 C 中用双引号引起来的若干个字符称为字符串常量。

5. 指针常量

任何类型的指针都可以赋以 0 值(NULL)，称为空指针，表示当前该指针不指向任何类型的任何变量，而不是指向地址为 0 的存储单元。因此，NULL 是一个指针常量。

6. 在 C 语言中，所有的变量都必须先定义后使用

其意义在于明确了变量的 4 个属性：变量的名字、数据类型、允许的取值范围以及合法的操作。定义变量的格式：

类型说明符 变量名表；

2.2.4 数据的输入输出

1. 程序要按照指定的格式输出，可调用格式输出函数 printf()，常用输出格式参见表 2-2 与表 2-3。调用printf()的一般格式为：

printf(格式控制字符串,输出项表);

表 2-2 常用的输出格式符

格式符	含 义	格式符	含 义
d 或 i	以十进制形式输出	e 或 E	以指数形式输出
o	以八进制形式输出	c	输出一个字符
x 或 X	以十六进制形式输出	s	输出字符串
u	以无符号十进制形式输出	%	输出字符
f	以小数形式输出		

表 2-3 常用的输出格式修饰符

修饰符	含 义
-	左对齐标志，默认值为右对齐
+	整数输出时带正号
#	输出八进制数时，前面加数字 0；输出十六进制数时，前面加数字 0x；输出浮点数时带小数点
数字	指定输出的宽度
. 数字	指定输出的小数点后的位数；对于 s 格式，指定输出的字符数
h	输出的是短整型
l	输出的是长整型
L	输出的是 long double 浮点数

2. 要按指定格式组织输入，可调用格式输入函数 scanf()，常用输入格式参见表 2-4 与表 2-5。调用 scanf()的一般格式为：

scanf(格式控制字符串,数据存储地址项表);

表 2-4 常用的输入格式符

格式符	含 义	格式符	含 义
d 或 i	以十进制形式输入	f 或 e	以指数形式输入
o	以八进制形式输入	c	输入一个字符
x 或 X	以十六进制形式输入	s	输入字符串
u	以无符号十进制形式输入		

表 2-5 常用的输入格式修饰符

修饰符	含 义
*	输入当前数据,但不传送给变量,相应格式也不对应存储地址
数字	指定输入数据的字符个数
h	输入短整型
l	输入长整型
L	输入 long double 浮点数

3. 要输出单个字符可调用字符输出函数 putchar()。putchar()函数的作用是将一个字符输出到标准输出设备上,这个设备通常指显示器。调用 putchar()的一般格式为:

```
putchar(ch);
```

4. 要输入单个字符可调用字符输入函数 getchar()。getchar()函数的作用是从标准输入设备上读入一个字符,这个设备通常指键盘。调用 getchar()的一般格式为:

```
getchar();
```

2.2.5 运算符

1. 赋值运算

1) 普通赋值运算。

变量名 = 表达式;

2) 复合赋值运算。

把赋值运算符与算术运算符结合在一起,构成 5 个算术复合运算符, +=、-=、*=、/=、%=。另外,把赋值运算符与移位和位逻辑运算符结合在一起,构成 5 个位运算的复合运算符, <<=、>>=、&=、^=、|=。

2. 算术运算

算术运算与数学上的数值运算很相似,是 C/C++ 中最简单和最容易理解的运算符。算术运算按运算分量划分,可分为单目运算符和双目运算符两类:

单目运算符: +(取正)、-(取负)

双目运算符: + 加法

- 减法

* 乘法

/ 除法

% 求余

3. 关系运算

关系运算就是比较两个数值大小的运算,比较两个数值大小的运算符就是关系运算符。一共有 6 个关系运算符,如表 2-6 所示。

表 2-6 关系运算符

运算符	<	<=	>	>=	==	!=
功能	小于	小于等于	大于	大于等于	等于	不等于

4. 逻辑运算

逻辑运算用于判断分析，逻辑运算的操作数和运算结果均为逻辑值。一共有 3 个逻辑运算符，如表 2-7 所示。

表 2-7 逻辑运算符

运算符	名 称	功 能
!	逻辑非，单目	操作数的值为真时，结果为假；操作数的值为假时，结果为真
&&	逻辑与，双目	当两个操作数全为真时，结果为真，否则为假
	逻辑或，双目	当两个操作数中只要有一个为真时，结果为真，否则为假

5. 条件运算

条件运算是 C 语言中唯一一个三目运算，条件运算的运算符为(?:)，需要三个操作数。条件运算的语法格式为：

表达式 1 ? 表达式 2 : 表达式 3

6. 自增/自减运算

自增/自减运算符是 C 语言最有特色的运算符。这两个运算符是：++ 和 --。

这两个运算符有以下四种表示方式：

++i 前缀形式：表示在引用变量之前，先使 i 加 1，以加 1 后的 i 值为运算结果。

--i 前缀形式：表示在引用变量之前，先使 i 减 1，以减 1 后的 i 值为运算结果。

i++ 后缀形式：表示在引用变量之后，才使 i 加 1，即以加 1 前的 i 值为运算结果。

i-- 后缀形式：表示在引用变量之后，才使 i 减 1，即以减 1 前的 i 值为运算结果。

2.2.6 表达式

由于 C 语言提供的运算符种类繁多，所以组成的表达式的种类也很多。常见的表达式有以下 6 种：

算术表达式：由算术运算符和圆括号将操作数连接起来的式子是算术表达式。

关系表达式：由关系运算符(>、<、>=、<=、==、!=)连接起来的表达式是关系表达式。

逻辑表达式：由逻辑运算符(&&、||、!)连接起来的表达式是逻辑表达式。

赋值表达式：由赋值运算符(=)连接起来的表达式是赋值表达式，它首先计算赋值号右边的表达式，然后将结果存入赋值号左边的变量名指定的存储单元中。

条件表达式：由条件运算符(?:)连接起来的表达式是条件表达式。

逗号表达式：由逗号运算符将两个表达式分开，形成逗号表达式。

2.3 上机实训

1. 实验目的

- 1) 掌握 C 语言基本标识符的写法，能正确地命名变量和文件。
- 2) 掌握 C 语言数据类型的基本概念，能够正确地定义整型、字符型、实型变量。
- 3) 掌握 C 语言运算符的规则，能够正确地运用 C 语言的各种运算符构成合法表达式。
- 4) 掌握不同数据类型混合运算时，数据类型的转换规则。
- 5) 掌握 C 语言数据、字符的输入输出方法。

2. 实验内容

- 1) 分析下面的程序，并上机验证结果。

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