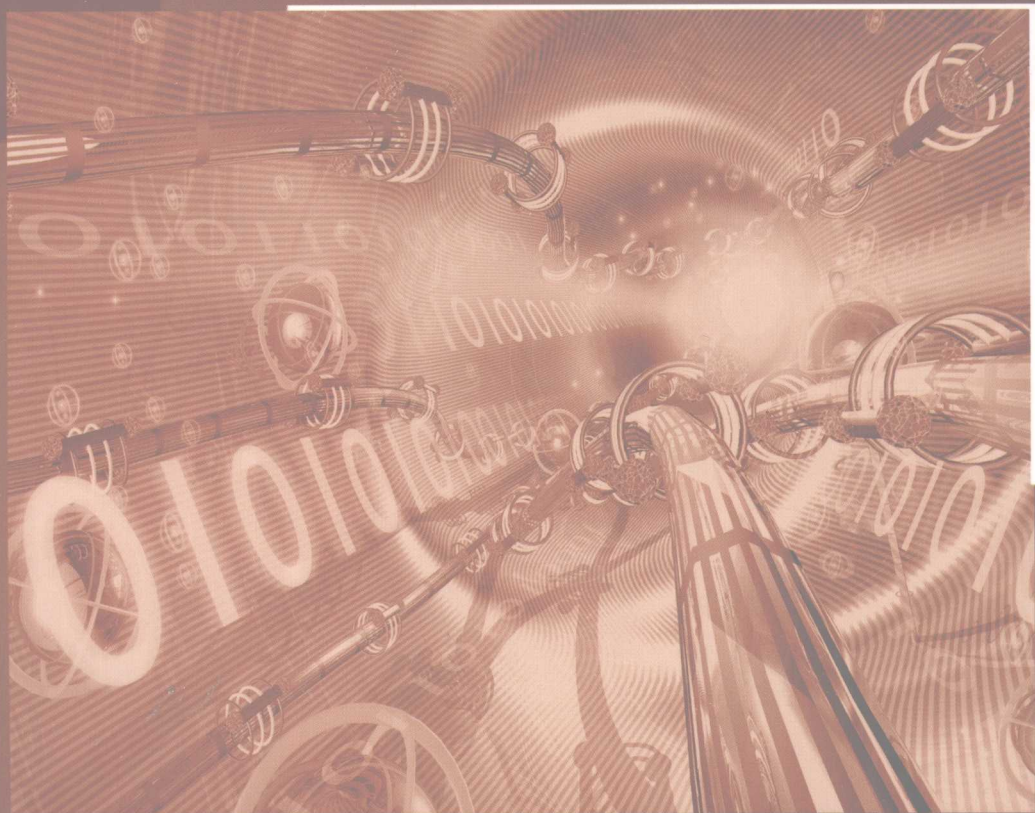


# 程序设计基础(C语言版) 学习与实验指导



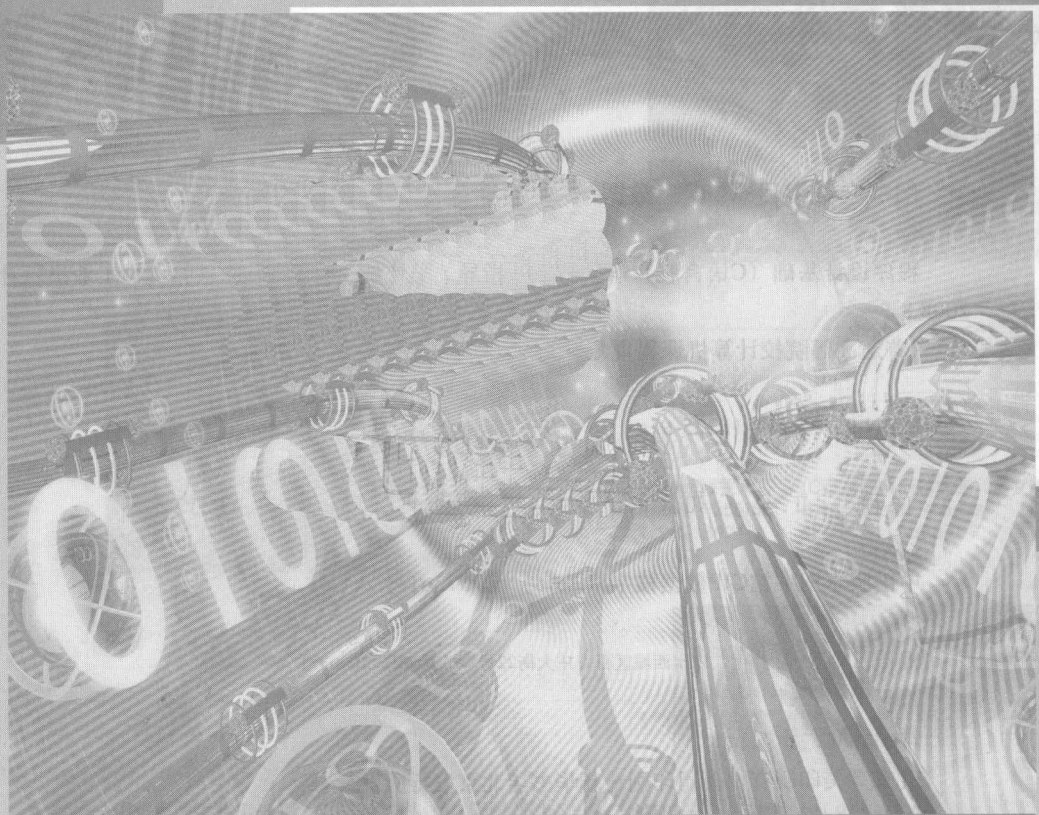
吴伟敏 秦军 林巧民 何丽萍 编著



机械工业出版社  
China Machine Press

普通高等院校计算机课程规划教材

# 程序设计基础(C语言版) 学习与实验指导



吴伟敏 秦军 林巧民 何丽萍 编著



机械工业出版社  
China Machine Press

本书是C语言教学的配套辅助教材。在内容的安排和组织结构上既考虑初学者，也兼顾希望深入了解C语言的读者。本书强调实用性，突出程序设计的方法和技巧，帮助学习者学会用计算机编程的思想考虑问题，学会用C语言解决实际问题。通过练习和上机实践掌握基本概念和基本算法，同时兼顾学科的前沿概念与技术。

全书分为两部分，第一部分是学习指导，通过知识要点、例题解析和习题及参考答案，为学习过程提供支持和帮助；第二部分为实验指导，共包括12个实验，涉及熟悉上机环境并编写简单程序、选择结构程序设计、循环结构编程、函数编程设计、高级函数编程、数组编程、指针编程、字符串编程、数组、指针和函数综合编程、结构体编程、文件编程和综合实验。

本书适合作为大学本科相关专业课程的配套辅助教材，对于自学者也是一本有价值的参考书。

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

程序设计基础 (C语言版) 学习与实验指导 / 吴伟敏等编著. —北京: 机械工业出版社, 2008.8

(普通高等院校计算机课程规划教材)

ISBN 978-7-111-11225-9

I. 程… II. 吴… III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字 (2008) 第057361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: 姚 蕾

责任编辑: 王 玉

北京慧美印刷有限公司印刷·新华书店北京发行所发行

2008年8月第1版第1次印刷

184mm×260mm·11.5印张

标准书号: ISBN 978-7-111-11225-9

定价: 20.00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

在目前常用的几种高级计算机语言中，C语言由于具有良好的结构化特性、简练的语法、强大的功能等众多优点，已被我国大多数高校选作高级语言的教学语言。学生在学习C语言的过程中，往往面临上课及看书时懂，作业也能做，但自己编程时却无从下手的情况。这个问题的根源在于学生对程序设计的基本方法和基本思想未能很好地掌握，未达到灵活应用的水平。对于教学过程中学生经常提出的“如何才能学好程序设计？”这样的问题，最好的回答就是多实践。本书正是依据这个前提，针对学生在实践过程中可能碰到的理论和操作上的难点和疑问，有的放矢地进行组织的。

在学习计算机语言课程时，不但应学习有关的理论知识、基本概念，更应注重实际的操作能力。学生通过上机操作，既可以加深对教材基本概念的理解，又可以提高动手操作的能力。只有学以致用、在运用中学习，才能善于运用所学知识解决实际问题，才能学得更扎实、深入，才能学有所成。

本书是《程序设计基础（C语言版）》（ISBN: 978-7-111-21975-0）的配套教学用书，也可作为高校各专业C语言的教辅教材和学习参考书。

全书分为两部分，第一部分为“学习指导”，与配套教材的章节一一对应，内容包括：C语言程序设计基本概念、数据类型、运算符与表达式、顺序结构程序设计、选择结构程序设计、循环结构程序设计、函数、编译预处理命令、数组、指针、字符串及其操作、结构体与共同体类型、文件处理和面向对象C++语言概述。其中的“知识要点”将对应章节的重点和难点进行归纳和总结，起到总结和强化的作用。“例题解析”部分对典型例题进行剖析，具有极强的针对性，所讨论的问题为教学过程中学生经常碰到的热点问题，彰显实用性，可以帮助读者掌握C程序设计的方法和技巧。“习题及参考答案”为精选的习题。这些习题在“例题解析”的基础上有所变化，让读者通过自行练习来巩固课堂上所学习的知识。

本书的第二部分为“实验指导”，共包括12个实验，内容涉及熟悉上机环境并编写简单程序、选择结构程序设计、循环结构编程、函数编程设计、高级函数编程、数组编程、指针编程、字符串编程、数组、指针和函数综合编程、结构体编程、文件编程和综合实验。各实验紧密配合教材内容，根据教学的需要，按照各章节的特点设置。实验部分所采用的例题专门针对学生在学习过程中容易混淆的知识点，提供多个综合性实验，帮助学生将分散的知识点融会贯通，并提高综合应用能力。每个实验都给出实验目的和要求，说明了实验后所能达到的水平。“实验准备”部分介绍相关的基础知识要点。“实验内容”部分明确实验的过程和步骤。“实验指导”部分指出实验内容中需要注意的事项和值得进一步探索的问题。“思考题”部分帮助学生在实验完成后对实验内容进行总结和升华。“选做题”为部分有能力的学生更好地掌握本实验相关内容提供深入探究的空间。

读者不但应掌握教材中的例题，也应对本书中的习题、例题以及实验一一上机操作。本书的题目类型多样，有些例题给出了全部程序，但并不是唯一的答案。读者可以考虑其他的方法，也可以对给出的程序进行补充，使程序更加完善。有些题目的程序抽出了一些语句，让读者填充，读者应在充分理解程序的基础上正确地填写，而且答案可能不止一种。还有些题目只给出了基本框架，读者可自行发挥。本书中实例已在Visual C++ 6.0环境下运行成功，读者可上机验证，或者进行改造，以便更好地理解相应的编程知识。

此外，附录A是根据以往教学实践中初学者常犯的错误提炼而成的，可以方便初学者在刚接触C语言时了解并避免这些错误。附录B可以帮助学生书写规范的实验报告。

本书的“学习指导”部分第7~10章以及附录A和附录B由吴伟敏编写，第1、11、12章由秦军编写，第2~5章由何丽萍编写，第6、13章由林巧民编写。本书的“实验指导”部分实验6~9、12由吴伟敏编写，实验10、11由秦军编写，实验1~3由何丽萍编写，实验4、5由林巧民编写。

南京大学徐浩磐教授在本书的编写过程中给予了具体指导和帮助，并担任主审，为本书提出了很多宝贵的意见，在此表示衷心感谢。

由于编者水平有限，书中的错误在所难免，恳请读者批评指正。

作者的电子信箱是wwm@njupt.edu.cn。

编者

2008年6月

## 目 录

前言	1	第8章 数组	58
第一部分 学习指导		8.1 知识要点	58
第1章 C语言程序设计基本概念	1	8.2 例题解析	59
1.1 知识要点	1	8.3 习题及参考答案	61
1.2 例题解析	2	第9章 指针	65
1.3 习题及参考答案	3	9.1 知识要点	65
第2章 数据类型、运算符与表达式	4	9.2 例题解析	67
2.1 知识要点	4	9.3 习题及参考答案	70
2.2 例题解析	7	第10章 字符串及其操作	76
2.3 习题及参考答案	10	10.1 知识要点	76
第3章 顺序结构程序设计	13	10.2 例题解析	78
3.1 知识要点	13	10.3 习题及参考答案	80
3.2 例题解析	15	第11章 结构体与共同体类型	92
3.3 习题及参考答案	19	11.1 知识要点	92
第4章 选择结构程序设计	22	11.2 例题解析	94
4.1 知识要点	22	11.3 习题及参考答案	97
4.2 例题解析	25	第12章 文件处理	102
4.3 习题及参考答案	27	12.1 知识要点	102
第5章 循环结构程序设计	32	12.2 例题解析	103
5.1 知识要点	32	12.3 习题及参考答案	104
5.2 例题解析	34	第13章 面向对象C++语言概述	109
5.3 习题及参考答案	37	13.1 知识要点	109
第6章 函数	41	13.2 例题解析	112
6.1 知识要点	41	13.3 习题及参考答案	113
6.2 例题解析	43	第二部分 实 验	
6.3 习题及参考答案	46	实验1 熟悉上机环境并编写简单程序	117
第7章 编译预处理命令	52	实验2 选择结构程序设计	123
7.1 知识要点	52	实验3 循环结构编程	126
7.2 例题解析	52	实验4 函数编程设计	131
7.3 习题及参考答案	54	实验5 高级函数编程	135



# 第一部分 学习指导

## 第1章 C语言程序设计基本概念

### 1.1 知识要点

C程序设计语言是一种结构化语言，诞生于1972年，目前仍广泛应用于各个领域。本章主要讲解C语言的基本概念。

#### 1. C语言的特点

C语言的优点主要如下：(1) 语言简洁紧凑，使用灵活方便；(2) 具有丰富的运算符；(3) 具有丰富的数据结构类型；(4) 生成的代码质量高；(5) 可移植性强；(6) 是结构化语言。

#### 2. 什么是程序设计语言

所谓计算机语言也就是计算机能识别的语言，通常称为程序设计语言。目前，程序设计语言大致可分为三类：机器语言、汇编语言和高级语言。C语言是一种高级语言。

#### 3. C语言的程序结构

C语言程序由一个或多个函数组成。一个C程序中必须有一个也只能有一个主函数，主函数的名字为main。除了主函数main，C程序还可以包括若干其他函数，函数由函数头部和函数体两部分组成。C程序的语句以分号为结束符。

下面是一段完整的C语言代码：

```
//C程序示例
#include<stdio.h>
void main()
{
    printf("Welcome to C programming\n");
}
```

#### 4. C程序生命周期（编辑、编译、链接和执行）

首先用编辑器，输入C语言源程序（编辑）。然后，利用编译器将源程序编译成目标程序（编译）。利用连接器将目标程序和库函数连接之后，生成可执行文件（链接）。最后，由操作系统执行该文件（执行）。

在C语言的整个生命周期中，有三类错误：在编译和连接时发生的语法错误，在运行时发生的运行错误和逻辑错误。

#### 5. 算法

任何计算问题的答案都是按指定的顺序执行一系列动作的结果，按照执行的动作和动作执行的顺序解决问题的过程称为“算法”。

一般，程序=数据结构+算法。数据结构是对数据的描述，指定数据的类型和数据的组织形式。算法是对操作的描述。现在很多专家对该公式进行了扩充：程序=数据结构+算法+程序设计方法+语言工具和环境。

算法有以下特性：确定性、有穷性、可行性、输入和输出。



算法的表示通常有自然语言、伪码、流程图和N-S图。

算法仅提供解决某类问题可采用的方法和步骤，还必须使用某种计算机程序设计语言根据算法的描述去实现。

## 6. 结构化程序设计方法

结构化程序设计的基本方法是：在设计程序时，本着自顶向下、逐步求精的原则，将一个复杂的大问题分解为若干独立的小问题（小模块）。每一个模块只有一个入口和一个出口。通过入口和出口将各个模块连接起来，通过多层次逐步求精，最后确定算法。结构化方法是自顶向下的程序设计方法。

结构化程序设计有三种基本控制结构，即顺序结构、选择结构和循环结构。

## 1.2 例题解析

【例1-1】C语言的基本单位是\_\_\_\_\_。

- A. 语句                      B. 程序行                      C. 函数                      D. 字符

解析：C

C语言程序由一个或多个函数组成。一个C程序中必须有一个，也只能有一个主函数，主函数的名字为main。除了主函数main，C程序还可以包括若干其他函数。

【例1-2】以下说法不正确的是\_\_\_\_\_。

- A. C程序中一行可以写多条语句                      B. C程序中的语句可以采用缩进形式  
C. C程序中每一行只能写一条语句                      D. C程序可以用注释提高程序的可读性

解析：C

C程序的书写风格比较自由，每条语句可以从任一列开始；每一行可以写多条语句，一条语句可以写在多行上；可以适当添加注释信息，提高程序的可读性。

【例1-3】C语言的结构特点是什么？

解析：C语言是一种结构化语言，C语言由一个主函数（main()函数）以及一个或多个函数所组成，其函数结构便于把一个大型的程序划分为若干相对独立的模块，模块间通过函数调用来实现相互连接，从而使函数模块化。

【例1-4】什么是结构化程序设计？

解析：将一个复杂的大问题分解为若干独立的小问题（小模块）。每一个模块只有一个入口和一个出口。通过入口和出口将各个模块连接起来，自顶向下，通过多层次逐步求精，最后确定算法的方法称为结构化程序设计。

【例1-5】调试下面的C程序，找出其中的错误并改正：

```
#include<stdio.h>
void main()
{ int a, b
  a=10;
  b=a*3;
  printf("B= %d\n", b);
}
```

解析：C语言程序的语句以分号为结束符。上面程序的第三行结束部分缺少分号。正确的程序如下：

```
#include<stdio.h>
void main()
{ int a, b;
  a=10;
  b=a*3;
  printf("B= %d\n", b);
}
```

### 1.3 习题及参考答案

#### 一、选择题

1. 在一个C语言程序中\_\_\_\_\_。  
A. main函数必须出现在所有函数之前  
B. main函数必须出现在所有函数之后  
C. main函数必须出现在固定位置  
D. main函数可以在任何地方出现
2. 以下叙述中正确的是\_\_\_\_\_。  
A. 构成C程序的基本单位是函数  
B. 一个完整的C程序中可以有main()函数  
C. main()函数必须放在所有函数之前  
D. C程序中只能有一个main()函数且不能再有其他的函数
3. 下列叙述中正确的是\_\_\_\_\_。  
A. C语言编译时不检查语法  
B. C语言中所有的函数都是系统提供的  
C. C程序中至少包含一个main()函数  
D. C程序不可任意添加注释

#### 二、填空题

1. 函数体由\_\_\_\_\_和\_\_\_\_\_组成。
2. 结构化程序由\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_三种基本结构组成。
3. 一个C程序的开发步骤包括程序编辑、程序编译、\_\_\_\_\_和程序运行。

#### 三、编程题

写一段程序，在屏幕上输出如下信息：

```
*****
*                LANGUAGE    C                *
*                BEGIN                *
*****
```

#### 四、参考答案

##### 选择题

1. D 2. A 3. C

##### 填空题

1. 说明部分 执行部分
2. 顺序结构 选择结构 循环结构
3. 连接程序

##### 编程题

```
#include <stdio.h>
void main()
{
    printf("*****\n");
    printf("*                LANGUAGE    C                *\n");
    printf("*                BEGIN                *\n");
    printf("*****\n");
}
```

## 第2章 数据类型、运算符与表达式

### 2.1 知识要点

数据是计算机程序在运行时的处理对象。数据类型决定了数据的运算范围、处理方式和存储表示方式。对数据进行处理时，C语言提供了非常丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算功能。本章将重点复习基本的数据类型、运算符及其相关的表达式。

#### 1. C语言的字符集和词法约定

C语言使用的词汇分为六类：标识符、关键字、运算符、分隔符、常量和注释符。

在程序中使用的变量名、函数名、标号等统称为标识符。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，标识符只能是字母、数字和下划线，并且第一个字符必须是字母或下划线。

在使用标识符时需要注意以下几点：

- 1) 标识符命名应尽量做到见名知意。
- 2) C语言的标识符区分大小写字母，即大写和小写分别代表不同的变量。
- 3) 标识符的长度受不同版本的C语言编译系统以及具体计算机系统的限制。

关键字是由C语言规定的具有特定意义的字符串，通常称为保留字。用户定义的标识符不应与关键字相同。

C语言中采用的分隔符有逗号和空格。

C语言的注释符是以“/\*”开头并以“\*/”结尾的串。程序编译时，对注释不作任何处理。

#### 2. 常量和变量

在程序运行过程中，其值不发生改变的量称为常量。从使用形式上看，常量包括字面常量和符号常量。直接以数据值表示的常量称为字面常量或直接常量；以标识符号（以下简称标识符）表示的常量称为符号常量。

在程序运行过程中值可以改变的量称为变量。学习变量，要区别开变量名和变量值以及变量的地址这三个概念。

#### 3. 数据类型

所谓数据类型是按数据的运算范围、处理方式、占据存储空间的大小，以及构造特点来划分的。在C语言中，数据类型可分为基本数据类型、构造数据类型、指针类型、空类型四大类，具体还可以再细分，如图2-1所示。

C语言的语法对于各种数据类型规定了不同的书写格式和操作规则，以及如何创建和使用变量来存放它们。

##### (1) 整型数据

1) 整型常量即表示整数的常量，C语言对整数有3种表示形式：十进制形式、以数字0开头的八进制形式和以0x（或0X）开头的十六进制形式。

2) 整型变量分为六类。各种无符号类型整型变量所占的内存空间字节数与相应的有符号类型变量相同，由于无符号类型无符号位，所以两者范围不同。各类型整型变量占用的内存

字节数随系统而异，表2-1给出ANSI标准定义的类型及其对应的字节数和最小取值范围。



图2-1 C语言数据类型

表2-1 ANSI标准定义的基本数据类型

数据类型	所占字节数	最小取值范围
有符号基本整型 (int)	2	-32768 ~ 32767 ( $-2^{15} \sim 2^{15}-1$ )
有符号短整型 (short int)	2	-32768 ~ 32767 ( $-2^{15} \sim 2^{15}-1$ )
有符号长整型 (long int)	4	-2147483648 ~ 2147483647 ( $-2^{31} \sim 2^{31}-1$ )
无符号基本整型 (unsigned)	2	0 ~ 65535 ( $0 \sim 2^{16}-1$ )
无符号短整型 (unsigned short)	2	0 ~ 65535 ( $0 \sim 2^{16}-1$ )
无符号长整型 (unsigned long)	4	0 ~ 4294967295 ( $0 \sim 2^{32}-1$ )

### (2) 实型数据

在C语言中，实型常量有十进制数形式和指数形式。实型变量分为单精度型和双精度型。表2-2给出了常用的C编译系统中对应的字节数、有效数字位和取值范围。

表2-2 数据类型

数据类型	所占字节数	有效数字位	取值范围
单精度型 (float)	4	6~7	$-3.4E+38 \sim 3.4E+38$
双精度型 (double)	8	15~16	$-1.7E+308 \sim 1.7E+308$
长双精度型 (long double)	16	18~19	$-1.2E+4932 \sim 1.2E+4932$

### (3) 字符型数据

字符型常量有两种表示方法，用单引号括起的单个字符这种普通的表示形式和转义字符这种特殊的表示形式。

字符型变量是用以保存字符类型数据的变量。字符型变量的类型说明符为char，只占一个字节的内存空间。

字符型数据实际保存对应字符的ASCII码整数值，所以存储形式类似整型。基于字符数据的这个特点，它可以出现在允许整型变量参与的运算中。



在C语言中,用一对双引号括起来的零个或多个字符序列称为字符串。字符串常量在内存中存储时,系统会自动在字符串的末尾添加一个转义字符'\0'作为字符串结束标志。该结束标志在内存中占用一个字节。字符串的长度是指字符串的有效字符个数,不包括双引号以及字符串结束标志。

#### 4. 运算符及表达式

##### (1) 算术运算符和算术表达式

###### 1) 基本的算术运算符。

单目运算符包括+和-两个运算符,出现在运算量左边,实现取原值和原值取反的操作,运算对象的类型可为整型或实型。

双目运算符包括+(加)、-(减)、\*(乘)、/(除)、%(求余或取模)。

在使用这些基本的算术运算符时,需要注意以下几点:

- 参与运算的对象数。
- 参与运算对象的类型。求余数运算“%”要求两侧的操作对象均为整型数据。
- 运算结果的类型。C语言规定:两个整数相除,其结果也为整数,小数部分被舍弃。当任意一个操作数为实数时,结果为实数。

###### 2) 自增(++)、自减(--)运算符。

自增、自减运算符用于变量加1和减1。前缀运算是变量先自增(减)1,再使用变量;后缀运算是先使用原变量值,然后变量再自增(减)1。

##### (2) 赋值运算符和赋值表达式

在C语言中,用“=”表示赋值运算符,它的作用是将右侧表达式的值赋给左侧变量。在使用赋值运算符时,需要注意:左边只能取变量,不能为常量或表达式;右边可取变量、常量或任意表达式。

复合赋值运算符是由赋值运算符之前加一个双目运算符构成。

定义变量时,给变量赋予初值称为变量初始化。

##### (3) 逗号运算符和逗号表达式

在C语言中,可以用逗号运算符“,”把多个表达式连接起来,称为逗号表达式。其一般形式为:

表达式1,表达式2,表达式3,⋯,表达式n

逗号表达式的最终取值就是最后一个表达式的值。

#### 5. 运算符的优先级和结合性

对多个运算符组成的表达式进行求值运算时,需要根据运算符的优先级次序和结合性执行。一般规则是:在C语言表达式中,优先级较高的先于优先级较低的进行运算。例如,先乘除后加减,如果有括号,则先计算括号。如果一个运算对象(或称操作数)两侧运算符的优先级相同,则按C语言规定的结合方向(结合性)进行。

#### 6. C语言中的数据类型转换

##### (1) 各类数值型数据间混合运算时的类型转换规则

整型、实型、字符型数据间可以混合运算。在这种情况下,需要将不一致的数据类型转换成一致的数据类型,然后进行运算。为了保证运算精度,系统在运算时的转换规则是将存储长度较短的运算对象转换成存储长度较长的类型,然后再进行处理。这种转换是系统自动进行的,如图2-2所示。

纵向箭头表示必定进行的转换,如float型数据必先转换为double型数据,然后与其他操作数进行运算。与此类似,char型或short型数据必先转换为int型数据,然后进行运算。

横向箭头表示当运算对象为不同类型数据时的转换方向，如int型数据与unsigned型数据进行运算，int型转换为unsigned型后方可进行运算。int型数据与double型数据进行运算，int型直接转换为double型后进行运算，不能理解为先转换为unsigned int型，然后转换为long int型，最后再转换为double型。

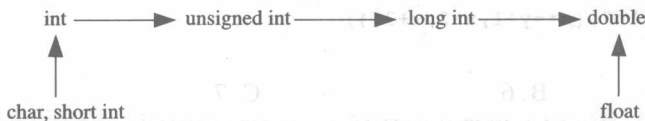


图2-2 混合运算时的类型转换规则

## (2) 赋值时的类型转换规则

当赋值运算符两侧的类型不同时，需进行类型转换，这种转换是系统自动进行的。

## (3) 强制类型转换

在C语言中，允许强制类型转换，将某一数据的数据类型转换为指定的另一种数据类型。强制转换是用强制转换运算符进行的，强制转换运算符为：(类型名)，强制转换运算符组成的运算表达式的一般形式为：(类型名)(表达式)。

强制转换运算符优先级比算术运算符高。与表达式中数据类型的自动转换一样，强制类型转换也是临时转换，对原运算对象的类型没有影响。

## 2.2 例题解析

【例2-1】下列可用于C语言用户标识符的一组是\_\_\_\_\_。

A. add-num, INT, Float

B. 08Olympiad, interger, n1\*n2

C. int, define, puts

D. sum, \_fun1, null

解析：答案为D。C语言规定，标识符只能由字母、数字、下划线组成，并且第一个字符必须是字母或下划线。所以排除选项A（有中划线）和B（第一个字符出现数字，另外出现字母、数字、下划线以外的字符）。C语言中还规定标识符不能为关键字，int 为关键字，所以排除选项C。

【例2-2】以下可作为C语言合法常量的是\_\_\_\_\_。

A. -0xde

B. -028

C. -e1.0

D. "

解析：答案为A。八进制数以0或-0开头，取值范围是0~7，所以选项B不合法。指数形式表示实型常量时，e前尾数一定要有数字，e后指数一定是整数，选项C不合法。字符常量是用单引号括起的单个字符，空格字符常量' '合法，但选项D只是两个连续的单引号，包含0个字符，不合法。而选项A中的de在十六进制数的合法取值范围0~f内，正确。

【例2-3】若有char c='2';，进行运算 c=c-'1'后 c的值为\_\_\_\_\_。

A. 不确定值

B. 1

C. 49

D. '1'

解析：答案为B。字符型变量保存的是字符数据的ASCII码值。对字符数据进行算术运算实际上是其ASCII码值参与运算，与整数的运算类似。字符数据'0'的ASCII值为48，后面的数据'1'到'9'依次加1。所以'2'-'1'的结果为两个相邻的ASCII值相减，结果为1，而不是'1'或'1'对应的ASCII值49。

【例2-4】C语言中下列运算符的操作数必须是int类型的运算是\_\_\_\_\_。

A. %

B. /

C. ++和--

D. %、++和--

解析：答案为B。C语言规定，求余数运算%要求两侧的操作对象均为整型数据。对相除运算/，两个整数相除，其结果也为整数，小数部分被舍弃。当任意一个操作数为实数时，结

果为实数。

【例2-5】下面程序的输出是 6。

```
#include<stdio.h>
void main( )
{int x=1,y=2;
  printf("%d\n", (x*=y+1,x*2,x+3));
}
```

- A. 9                      B. 6                      C. 7                      D. 5

解析：答案为B。在C语言提供的运算符中，逗号运算符的优先级最低，所以对逗号表达式  $(x*=y+1,x*2,x+3)$ ，先计算每个表达式，最终取值为最后一个表达式的值。另外，对  $x*=y+1$  这样的复合赋值运算符表达式，应理解为： $x=x*(y+1)$ 。根据以上原则，先计算出第一个表达式，值为3， $x$ 为3；第二个表达式值为6， $x$ 保持不变；第三个表达式值为6， $x$ 保持不变。

【例2-6】下面程序的输出是 9。

```
#include<stdio.h>
void main( )
{ int x = 011;
  printf( "%d\n", x++);
}
```

- A. 9                      B. 10                      C. 12                      D. 011

解析：答案为A。以数字0打头的数为八进制整数，所以执行“ $x = 011;$ ”后， $x$ 的值为十进制的9；C语言中的后缀自增运算符的运算规则是先使用原变量值，然后自增1，所以“ $\text{printf}(\text{" \%d\n"}, x++);$ ”语句先以十进制输出 $x$ ，然后 $x$ 增为10。

【例2-7】若有如下的变量定义：“ $\text{int } a, b; \text{ double } x=4, y;$ ”，则以下符合C语言语法的表达式是 A。

- A.  $a*=b/=a+++b$       B.  $a+b=2$       C.  $(\text{int})x\%(2++)$       D.  $y= \text{double}(a)$

解析：答案为A。赋值表达式要求左边只能取变量，不能为常量或表达式；右边可取变量、常量或任意表达式。C语言编译系统在处理表达式时一般从左到右扫描，将尽可能多的字符组成一个合法运算符，所以 $a+++b$ 等效于 $(a++)+b$ ，所以表达式 $a*=b/=a+++b$ 合法。 $a+b=2$ 的左边为表达式，所以为非法表达式。自增自减运算符的运算对象只能是变量，不能为常量或表达式，所以 $(\text{int})x\%(2++)$ 为非法表达式。强制转换运算符组成的运算表达式的一般形式为： $(\text{类型名})(\text{表达式})$ ，所以 $y= \text{double}(a)$ 的合法形式应为： $y=(\text{double})(a)$ 。

【例2-8】声明： $\text{int } i; \text{ float } x;$ ，则执行语句“ $x=i-1.6;$ ”后， $x$ 的值为 1.000000。

解析：答案为 $x=1.000000$ 。赋值运算符的结合性为右结合性，所以上述语句从右至左运算。根据C语言赋值时的类型转换原则， $\text{float}$ 、 $\text{double}$ 型数据赋值给 $\text{int}$ 型数据时，直接截断小数。 $\text{int}$ 、 $\text{char}$ 型数据赋值给 $\text{float}$ 、 $\text{double}$ 型数据时，补足有效位。所以第一步运算， $i$ 的值为1，第二步运算，1补足有效数字后赋值给 $x$ 。

【例2-9】请写出以下数学式的C语言表达式  $(5*\exp(m))/(x+\sin(30*3.1415926/180))$ 。

$$\frac{5e^m}{x+\sin 30^\circ}$$

解析：答案为 $(5*\exp(m))/(x+\sin(30*3.1415926/180))$ 。这个表达式需要使用C语言的数学库函数。

【例2-10】C语言中，字符常量“ $\backslash n$ ”、“ $\backslash 101$ ”、“ $\backslash x31$ ”称作 转义字符，其中“ $\backslash n$ ”表示 换行。

'\01'表示 \_\_\_\_\_, 字符串"\t\123\xee"的串长为 \_\_\_\_\_, 占内存空间 \_\_\_\_\_ 个字节。

解析: 答案为转义字符 换行 'A' 5 6。字符常量有一种特殊的表示形式, 称为转义字符。例如, '\ '表示一个反斜杠字符, '\t'代表一个水平制表位, '\"表示一个双引号, '\ddd'和'\xhh'分别用八进制数和十六进制数表示一个字符的ASCII码。字符串常量在内存中存储时, 系统会自动在字符串的末尾添加一个转义字符'\0'作为字符串结束标志。该结束标志在内存中占用一个字节。字符串的长度是指字符串的有效字符个数, 不包括双引号以及字符串结束标志。

【例2-11】下面程序的运行结果是 \_\_\_\_\_。

```
#include<stdio.h>
void main()
{ short n=32768;
  unsigned short m=n;
  printf("n=%d,m=%d",n,m);
}
```

解析: 答案为 $n = -32768$ ,  $m = 32768$ 。此题考察C语言编译程序对有符号/无符号整型数据的处理以及赋值转换原则。unsigned int型赋值给int型或非unsigned数据类型赋值给位数相同的unsigned 数据, 原则是直接传送数值。数据在计算机中是用二进制存储的, 所以, 32768的二进制数( $2^{15}$ )先直接传给m和n。根据一个整数是有符号数还是无符号数, C语言编译程序对其最高位的理解有区别。对于有符号整数n, 最高位代表符号位。所以 $2^{15}$ 是一个负数的补码, 其原码为-32768、对于无符号整数m, 所有位数包括最高位都作为数值位, 所以,  $2^{15}$ 就是m的原码值。

【例2-12】下面程序的运行结果是 \_\_\_\_\_。

```
#include<stdio.h>
void main()
{ char c=0xff80;
  printf("%d",--c);
}
```

解析: 答案为127。0xff80为2字节整数常量, char型数据c所占内存空间为1个字节, 赋值时, c取右边常量的低8位, 结果为10000000。以十进制输出时, 先执行自减运算, 10000000和-1的补码11111111进行加法运算, 最高位溢出, 结果为01111111。

【例2-13】请分析以下程序有什么问题并给出改正方法。

```
#include<stdio.h>
void main()
{ char c1="1";
}
```

解析: 改正方法为把赋值号右侧的字符串常量改为字符常量: `char c1='1'`;。char类型的变量占一个字节内存空间, 字符串"1"在内存中占两个字节空间。所以编译时会给出warning警告。

【例2-14】以下程序是求三个数之和, 请分析程序有什么问题并给出改正方法。

```
void main()
{ int n1=n2=n3=1;
  sum+=n1+n2+n3;
}
```

解析: 1) 改正方法为: `int n1,n2,n3; n1=n2=n3=1;`或者`int n1=1,n2=1,n3=1;`。变量初始化与赋值语句赋值是不同的概念, 允许对已定义的变量用一个值同时赋值,



但不允许用一个值在定义变量时同时初始化它们。

2) 改正方法为: 添加变量sum的定义。变量要先定义后使用。在本程序中, sum变量没有定义就开始使用;

```
void main()
{ int n1,n2,n3, sum;
  n1=n2=n3=1;
  sum+=n1+n2+n3;
}
```

3) 上述改正后的程序仍有问题。改正方法: 定义时初始化为0或用赋值语句赋值为0。使用变量时的一个好习惯应该是初始化或赋值后再使用。虽然添加了sum变量的定义, 因为没有初始化或赋值, sum中的值不确定。如果求和语句为sum=n1+n2+n3, 运行时的结果正确, 但本程序是在sum原值上累积, 运行时得到的结果不是期望值。

完整的答案:

```
void main()
{ int n1,n2,n3,sum=0; //求和清0。如果是求积,则置1
  n1=n2=n3=1;
  sum+=(n1+n2+n3); //或为 sum=n1+n2+n3;
}
```

## 2.3 习题及参考答案

### 一、选择题

- 若变量a是int类型, 并执行了语句: a='A'+1.6;, 则正确的叙述是: \_\_\_\_\_。
  - a的值是字符C
  - a的值是浮点型
  - 不允许字符型和浮点型相加
  - a的值是字符'A'的ASCII值加上1
- 若已定义x和y为double类型, 则表达式: x=1, y=x+3/2的值是 \_\_\_\_\_。
  - 1
  - 2
  - 2.0
  - 2.5
- 若定义: int a=8, b=5, c;, 则执行语句“c=a/b+0.4;”后, c的值为 \_\_\_\_\_。
  - 1.4
  - 1
  - 2.0
  - 2
- 若有以下定义: char a; int b; float c; double d; 则表达式a\*b+d-c值的类型为 \_\_\_\_\_。
  - float
  - int
  - char
  - double
- 若int类型占两个字节, 则以下语句输出为 \_\_\_\_\_。
 

```
int k=-1; printf("%d,u%\n",k,k);
```

  - 1, -1
  - 1,65536
  - 1, 32768
  - 1, 65535
- 设int x=11;, 则表达式(x++ \* 1/3)的值是 \_\_\_\_\_。
  - 3
  - 4
  - 11
  - 12
- 设声明语句: char a='\72';, 则变量a \_\_\_\_\_。
  - 包含1个字符
  - 包含2个字符
  - 包含3个字符
  - 声明不合法
- 下列变量定义中合法的是 \_\_\_\_\_。
  - short \_a=1-1e-1;
  - double b=1+5e2.5;
  - long do=0xfdaL;
  - float 2\_and=1-e-3;
- 下列标识符中不合法的是 \_\_\_\_\_。
  - s\_name
  - \_e
  - math
  - 3DS
- 下列不正确的转义字符是 \_\_\_\_\_。
  - \"
  - \"
  - '074'
  - '\0'