新起点电脑教程

第 1 章

C 语言概述

本章要点

- 了解 C 语言的发展与特点
- 熟悉 C 语言开发环境
- 通过实例了解 C 程序的组成与格式

本章主要内容

C 语言是目前世界上最为流行的计算机高级程序设计语言之一。它设计精巧、功能齐全，既适合于编写应用软件，又适合于编写系统软件、目前广泛应用于底层开发。据统计，PC 机领域许多著名的系统软件和应用软件，都是运用 C 语言加上汇编语言子程序编写而成的。

1.1 C 语言的发展

C 语言是一种面向过程的语言，它介于低级语言和高级语言之间，同时且有高级语言和汇编语言的优点，可以广泛地应用于不同的操作系统，是目前应用最广泛的计算机语言之一。

↑扫码看视频

1.1.1 C 语言的历史

早期的计算机，都是用机器语言和汇编语言来编写程序代码。

1960 年开发的 ALGOL-60，对以后的高级语言的发展起到了很好的作用，但是语言过于抽象，没有得到推广。

1963 年，英国剑桥大学推出了 CPL(Combined Programming Language)语言。它比 ALGOL-60 更接近于硬件，但其规模较大，难以实现和学习。

1967 年，剑桥大学的 Martin Richards 对 CPL 语言进行了简化，于是产生了 BCPL(Basic Combined Programming Language)语言。

1970 年，美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础，设计出很简单且很接近硬件的 B 语言(取 BCPL 的首字母)，并且他用 B 语言写了第一个 UNIX 操作系统。

1972 年，美国贝尔实验室的 D. M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。

1973 年初，C 语言的主体完成。K. Thompson 和 D. M. Ritchie 两人合作，把原来用汇编语言编写的 UNIX 操作系统中 90%以上的代码用 C 语言来重写，即 UNIX 5。

随着 UNIX 的发展，C 语言自身也在不断地完善。直到今天，各种版本的 UNIX 内核和周边工具仍然使用 C 语言作为最主要的开发语言，其中还有不少继承于 Thompson 和 Ritchie 之手的代码。

1977 年，D. M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。

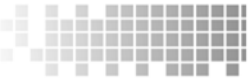
1978 年，BrianW. Kernighan 和 D.M.Ritchie(合称 K&R)合著了影响深远的 *The C Programming Language* 一书。该书中介绍的 C 语言被称为标准 C。

1982 年，美国国家标准协会成立了 C 标准委员会，用来建立 C 语言的标准。

1983 年，在参考 C 语言的各种版本的基础上，制定了新的标准，成为 ANSI C。

1988 年，K&R 按照 ANSI C 标准重写了 *The C Programming Language* 一书。

1989 年，ANSI 发布了第一个完整的 C 语言标准——ANSI X3.159–1989，简称 C89，人



们仍习惯称其为 ANSI C。

1990 年，国际标准化组织 ISO(International Standard Organization)接受了 C89 为 ISO C 的标准。ISO 官方给予的名称为 ISO/IEC 9899，简称 C90。 目前流行的 C 编译系统都是以它为基础的。

1999 年，在做了一些必要的修正和完善后，ISO 发布了新的 C 语言标准，命名为 ISO/IEC 9899:1999，简称 C99。

2011 年 12 月 8 日，ISO 又正式发布了新的标准，称为 ISO/IEC9899: 2011，简称 C11。它是 C 语言的第三个官方标准，也是 C 语言的最新标准，该标准更好地支持了汉字函数名和汉字标识符，一定程度上实现了汉字编程。

高级语言发展至今，面向对象的程序设计语言越来越受到人们的青睐，比如 Visual Basic(VB)、Visual C++(VC++)、C++、Java、C#等。其中，功能比较强大的还是 C++语言，它以 C 语言为基础，在很多方面两者兼容。因此，掌握了 C 语言，对进一步学习 C++或其他面向对象的程序设计语言如 Java、C#等会有非常大的帮助。

1.1.2　C 语言的特点

C 语言不仅有易学易用的优点，而且具有面向过程、结构化、可移植性强及集成开发环境先进等特点。而且，C 语言还具有很强的绘图能力、数据处理能力，同样适合于二维、三维图形和动画的开发。

C 语言之所以能够在众多的高级语言竞争中脱颖而出，成为高级语言中的佼佼者，主要是因为与普通高级语言相比，它具有以下特点。

1．是一个特殊的高级语言

C 语言把高级语言的基本结构和语句与低级语言的实用性结合起来，允许直接访问物理地址，对硬件进行操作。因此 C 语言既具有高级语言的功能，又具有低级语言的功能，能够像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元，因此可用来编写系统软件。因此也有人把 C 语言称为中级语言。

2．C 语言简洁紧凑，使用方便灵活

C 语言一共只有 32 个关键字，9 种控制语句，压缩了一切不必要的成分。程序书写形式自由，主要用小写字母表示。

3．运算符丰富，表达能力强

C 语言中有 44 个运算符。除了算术、关系、逻辑等常规运算符之外，还含有指针、地址、位、自增自减、条件、复合赋值运算符，甚至连圆括号、方括号、逗号、小数点都能够作为运算符。由于 C 语言的运算符、表达式类型极为丰富多样，所以能够实现各种各样的高级和低级的、其他高级语言难以实现的运算。

4．数据类型丰富

C 语言数据类型包括整型、实型、字符型、枚举类型，结构体、共用体、数组和文件类

型，指针类型，空类型。其中，整型、实型、字符型中还有多种小的类型。指针类型是 C 语言中最具特点的一种数据类型。它使用起来非常灵活，把 C 语言的功能特点发挥得淋漓尽致。

5．C 语言语法限制不太严格，程序设计自由度大

比如，数组下标不做超界检查，整型、字符型、逻辑型可以通用。这些程序设计的灵活性在一定程度上降低了安全性，所以也对程序设计人员提出了更高的要求。

6．生成的目标代码质量高

C 语言简洁、紧凑，程序执行速度快，可读性好，易于调试、修改和移植。它比一般的高级语言生成的目标代码质量高约 20%，只是比汇编语言低 10%～20%，在高级语言中是出类拔萃的。

7．可移植性好

C 语言在不同机器上的 C 编译程序，86%的代码是公共的，是通过调用系统提供的库函数来实现的，所以 C 语言的编译程序便于移植。在一个环境上用 C 语言编写的程序，不改动或稍加改动，就可移植到另一个完全不同的环境中运行。

目前，C 语言在其原有应用领域的基础上，又拓展了新的应用领域，支持大型数据库开发和 Internet 应用以及嵌入式系统的应用。

嵌入式系统体现了目前最新科技水平，也是当前最热门、最有发展前途的 IT 应用领域之一。要应对嵌入式系统技能特有的挑战，就要学习 C 语言的程序设计，目前，最为广泛应用的是嵌入式 C 语言编程、Linux C 语言编程等。

1.2 C 语言的开发环境

C 语言编程的开发环境有很多种，其中最常用的有 Turbo C 2.0 集成开发环境和 Visual C++6.0(简称 VC++)系统开发环境两种。

↑扫码看视频

1.2.1 Turbo C 2.0 集成开发环境

Turbo C 2.0 是 Borland 公司开发的一个 C 语言集成开发环境，以占用资源少、编译速度快、代码执行效率高而著称，也是 C 语言程序开发者最乐于使用的编程工具。

开发一个 C 语言程序的基本步骤可用图 1.1 描述。C 语言程序经过编辑、编译、链接、

生成 exe 可执行文件，然后在计算机上执行。无论哪个阶段有错误，都要回到编辑状态修改源程序。修改后再编辑、编译、链接、执行。经过编辑保存之后的源文件，默认的扩展名是.c，经过编译之后生成的文件扩展名为.obj，经过链接之后生成的文件扩展名为.exe。执行时，使用的是生成的 exe 文件。

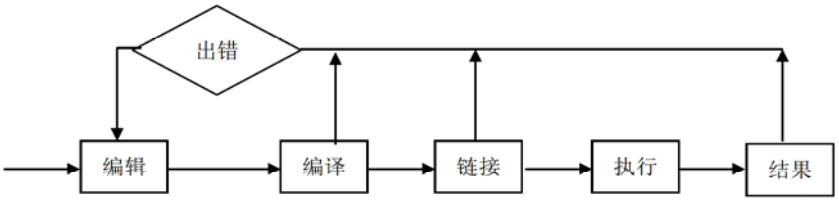


图 1.1　C 语言程序开发步骤

1. Turbo C 的界面

首先将 TC 编译程序保存在计算机磁盘的某一目录下，比如 C 盘的 TC 目录下。

双击打开文件 TC.exe，进入 Turbo C 主界面，如图 1.2 所示。如果界面不是全屏显示，可在 TC.exe 图标上右击，选择“属性”命令，再选择“屏幕”选项卡，单击“全屏幕”按钮即可把 TC 界面设为全屏。

图 1.2　C 语言主界面

界面顶部是主菜单项，按 F10 键可以随时激活主菜单。按←、→光标键可以选择主菜单项，按 Enter 键可以确认选中该菜单项。除 Edit 项外，其余菜单项都有下拉子菜单。有些菜单项存在着三级菜单。对于下拉子菜单，可以使用↑、↓光标键来选择。

其中 File 菜单是使用最多的一个菜单。File 菜单中各命令的功能说明如表 1.1 所示。

表 1.1　File 菜单中各命令的功能说明

命　令	功能说明
Load 或 F3 键	载入文件。提示输入一个文件名(可以是新文件)，把文件载入到编辑窗口
Pick 或 Alt+F3 键	选取文件。显示一个子菜单，列有编辑器最后装入过的 8 个文件名
New	新建一个文件。默认的文件名为 NONAME.C
Save 或 F2 键	保存文件。如果文件未命名，系统将等待输入一个新的文件名。再次保存将覆盖原来的文件内容

续表

命　令	功能说明
Write to	将文件更名存盘。系统将要求输入一个新的文件名，不会更改原来的文件内容
Directory	显示目录及所需文件列表(按 Enter 键选择当前目录)
Quit 或 Alt+X 键	退出 Turbo C 系统，回到 DOS 或进入 TC 前的状态。若再使用 Turbo C 系统，需要重新双击 TC.exe 进入

选择 Edit 菜单可以进入编辑窗编写程序。

用户第一次上机，需要对编程环境进行必要的设置。否则，编译时会显示语法错误。进入 Options 菜单中的第 4 项 Directories 命令，如图 1.3 所示。

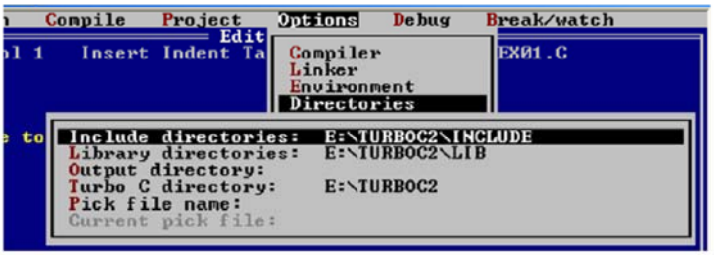


图 1.3　Options 选项

- 将 Include directories 项修改为 C:\TC\INCLUDE。
- 将 Library director 项修改为 C: \TC\LIB。
- 将 Output directory 项修改为 C: \TC。
- 将 Turbo C directory 项修改为 C: \TC。
- 修改完成后，按 Esc 键返回上一级目录，再选择 Save Options 命令保存修改。

2. Turbo C 的编辑操作

按 Alt+E 快捷键进入编辑状态，首行会提示正在进行编辑操作的信息：Line、Col 表示当前光标所在的行、列；Insert 表示编辑处于插入状态，可用 Insert 键切换；Indent 表示齿形自动缩进，可提高程序的可读性，按 Ctrl+O+I 快捷键可切换为 Unindent 不缩进状态；Tab 表示可插入制表符，可以用 Ctrl+O+T 快捷键切换。

编辑状态时可用←、→、↑、↓光标键上下左右移动光标，并在光标处进行插入、删除、修改等操作。

3. 最常用到的命令和快捷键

- F10：激活主菜单。
- F2：保存文件。
- F3：在编辑器中载入一个文件。
- Ctrl+F9：运行当前程序。
- Alt+F5：查看运行结果。即切换到用户屏幕，按任意键返回编辑窗口。
- F6：信息显示窗口与代码编辑窗口切换。

- F5：可使编辑窗口或信息窗口扩大到整个屏幕，按 F5 键可切换回来。
- F9：编译多个源文件构成的程序。
- Alt+X：退出 TC 环境，返回 DOS 状态。
- Esc：退出本级菜单，返回上级菜单。

熟练使用这些常用的功能键，可以提高程序的编辑与操作速度。

4. Turbo C 上机操作实例

下面以运行一个简单的 C 程序为例，演示上机操作过程。

启动 TC，进入 TC 环境，系统默认新建的文件名为 NONAME.C。按 Esc 键后，在编辑窗口中出现闪烁的编辑光标，就可以输入源程序的代码。

【例 1-1】输入下面的程序代码，每个语句结束后按 Enter 键。注意字母的大小写，标点符号用英文半角。

程序代码：

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

输入中若出现错误，可以用↑、↓、←、→键移动光标到错误文字符位置处进行修改。

输入结束后按 Ctrl+F9 快捷键，编译源程序并生成可执行文件。如程序没有错误，编译成功，则直接显示运行结果。按 Enter 键可返回编辑界面，如图 1.4 所示。

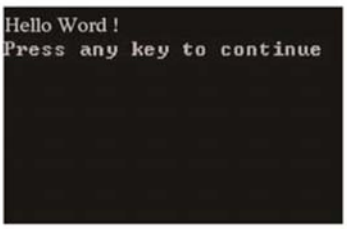


图 1.4　运行结果

如果当某行有语法错误时，编译将不会继续，系统会将该行白亮显示，并用一个竖的线条标出错误所在。可按 F6 键在代码编辑窗口和信息显示窗口之间进行切换，根据语法规则和错误提示进行修改，修改后重新按 Ctrl+F9 快捷键编译程序。

在编写源文件的过程中，可随时按 Alt+F5 快捷键查看运行结果。

1.2.2　Visual C++ 6.0 开发环境

Visual C++ 6.0 是一个功能强大的可视化软件开发工具，它将程序的代码编辑、程序编译、链接和调试功能集于一体。Visual C++ 6.0 集成环境运行于 Windows 平台，具有图形窗口界面，对于习惯使用鼠标的用户来说会感觉比较容易操作。

1. 安装与启动 Visual C++ 6.0

运行 Visual Studio 软件中的 setup.exe 程序，选择“安装 Visual C++ 6.0”，然后按照安装程序的向导，完成安装过程。

安装完成后，在“开始”菜单的“程序”中，选择“Microsoft Visual Studio 6.0”下的

“Microsoft Visual C++ 6.0”即可运行。也可以在 Windows 桌面建立一个快捷方式，双击即可进入 Visual C++ 6.0 运行环境。

2. 用 Visual C++ 6.0 建立并运行 C 语言应用程序

第 1 步 创建项目

用 Visual C++ 6.0 建立 C 语言应用程序，首先要创建一个工程(project)，用来存放 C 语言程序的所有信息。

进入 Visual C++ 6.0 环境后，选择主菜单“文件”中的“新建”命令，在弹出的对话框中打开“工程”选项卡，在列表框中选择 Win32 Console Application 工程类型，在“工程”文本框填写工程名，比如 exp，在“位置”文本框填写工程路径(即工程文件存放的位置，本例是在 C 盘建立一个名为 VC 的文件夹)，如图 1.5 所示。

图 1.5　建立工程文件

单击“确定”按钮之后，在弹出的 Win32 Console Application 窗口中默认选择 An empty project 选项，单击“确定”按钮，如图 1.6 所示。

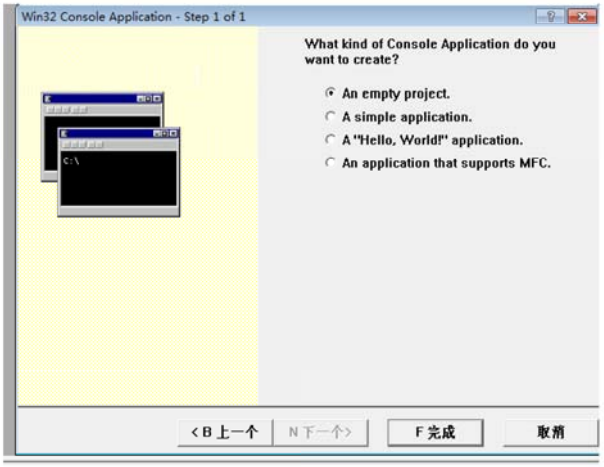


图 1.6　Win32 Console Application 窗口

在弹出的“新建工程信息”窗口中，直接单击“确定”按钮，即建立了一个新的工程项目。

第 2 步 创建 C 语言源程序文件

选择主菜单“文件”中的“新建”命令，在弹出的对话框中打开“文件”选项卡，在显示窗口选择 C++ Source File 文件类型，将文件名定义为 example.cpp，单击“确定”按钮，如图 1.7 所示。

图 1.7　新建 C 源文件

第 3 步 编写代码

在弹出的编辑窗口中输入源文件：

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

如图 1.8 所示。

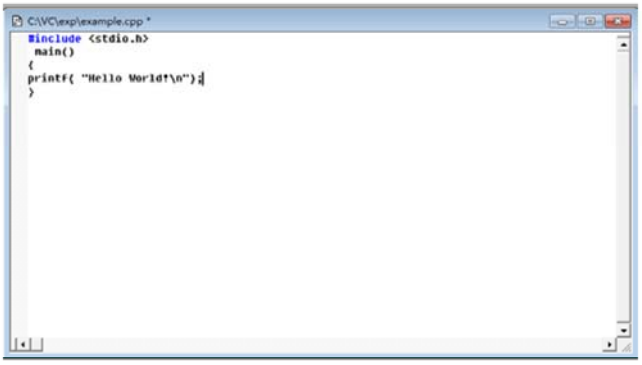


图 1.8　C 语言源文件

第 4 步 编译、链接和运行 C 语言源程序

C 语言源程序编辑完成后，选择主菜单“编译”中的“执行”命令，或单击工具栏上的

图标。

如果程序没有错误，将出现程序运行结果，如图 1.9 所示。

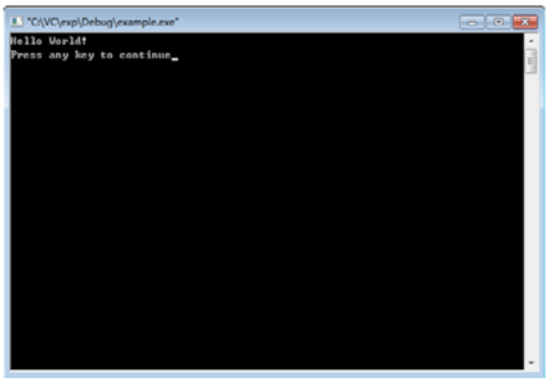


图 1.9　运行结果

第 5 步 检错

如果在编译过程中发现错误，将会在屏幕下方的显示错误与警告区域，显示所有的错误和警告信息。双击错误或警告的第一行，则光标自动跳到代码的错误行；修改错误后，重新进行编译，直到没有错误和警告信息为止。

1.3　简单 C 程序的组成和格式

本节将通过几个简单的例子，介绍 C 程序的一些基本概念，如函数、主函数、库函数等，并介绍 C 程序的基本编写要求。

↑扫码看视频

【例 1-2】在屏幕上输出字符：This is a C program.

程序代码：

```
#include <stdio.h>
 main()
{
    printf(" This is a C program.\n");              /*屏幕输出*/
}
```

运行结果：

```
This is a C program.
```

说明：(1)　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有一个主函数。函数中的语句放在一对花括号｛｝内，花括号括起来的部分称为函数体。“｛”表示函数体的开始，“｝”表示函数体的结束。小括号()内是主函数的参数，主函数可以没有参数，但是小括号不能省略。

(2)　C 语言中将需要实现的功能分别编写成若干函数，所谓函数名，就是给该功能起一个名字，主函数通过调用多个函数，组成一个完整的程序。函数之间也可以互相调用。函数是组成 C 程序的最基本单位。

(3)　函数体中可以有多个语句，每个语句以分号(；)结束。

(4)　本例中函数体中只有一个语句——printf 语句，它调用了库函数中的 printf()函数。printf()是打印函数，作用是在屏幕上输出双引号内的内容。其中“\n”是换行符，它的作用是在双引号中的内容输出完毕之后，将打印机头或屏幕光标换至下一行的起始位置。

(5)　#include 命令是编译预处理命令，也简称命令行，其作用是将后面引用的文件中的内容，读入到该语句位置处。

C 语言中数据的输入、输出函数都是通过调用系统提供的库函数 scanf()和 printf()等来实现，这些函数的说明都包括在 stdio.h 文件中。C 程序经常需要调用一些库函数来实现特定的功能，因此文件头需要加一些文件包含的命令行。库函数可以用“<　>”来引用，也可以用“"　"”来引用。

(6)　用/*…*/括起来的内容是语句的注释部分，仅供阅读程序，计算机并不执行注释语句的内容。“/*”必须成对出现，“/”和“*”之间不可以有空格。注释语句可以出现在程序中的任意位置。灵活使用注释语句可以增加程序的可读性。

【例 1-3】求输入的两个数的和。

程序代码：

```
#include <stdio.h>                    /*输入、输出函数包含的头文件*/
main()                                /*定义主函数*/
{
    int a, b, s;                      /*定义整形变量 a，b，s*/
    scanf("%d, %d", &a, &b);          /*键盘输入两个整数，放入变量 a 和 b 中*/
    s=a+b;                            /*将 a、b 求和后放入变量 s 中*)
    printf("s=%d\n", s);              /*%d 表示把 s 的值按十进制整型数据输出*/
}
```

运行结果：

```
10,20↙
s=30
```

说明：(1)　C 语言中大小写表示不同的含义，程序语句一般用小写字母书写，大写字母一般用做符号常量。

(2)　C 语言中使用的所有变量都必须先定义为某种数据类型，然后才能使用。

1.4 思考与练习

本章首先介绍了 C 语言的历史与发展、C 语言的特点及开发环境，并通过实例介绍了在 Turbo 2.0 及 Visual C++ 6.0 开发环境下如何使用 C 语言进行程序设计。通过本章的学习，读者应该能够使用 Turbo 2.0 及 Visual C++ 6.0 的开发环境编写简单的 C 语言程序。

一、简答

1. 简述 C 语言的主要特点。

2. 用 Turbo 2.0 编写，C 程序源文件的扩展名是什么？用 Visual C++ 6.0 编写，C 程序源文件的扩展名是什么？

3. C 语言程序的运行一般要经过哪几个步骤？

4. 构成 C 语言程序的基本单位是什么？它由哪几部分组成？

二、上机练习

1. 输入以下程序，得出运算结果。

```
#include<stdio.h>
main()
{
    int  x,y,s;
    x=50;
    y=300;
    s=x+y;
    printf("s=%d\n",s);
}
```

2. 输入以下程序，得出运算结果。

```
#include<stdio.h>
main()
{
    int x,y,z;
    x=10;
    y=20;
    z=x*y;
    printf("x=%d,y=%d\n",x,y);
    printf("z=%d\n",z);
}
```

三、编写程序

1. 打印如下图形。

```
*******
```

```
*****
 ***
  *
```

2．打印出如下图形。

```
**********************
I  love  C  programs!
**********************
```

3．给出圆的半径，计算圆的周长和面积。

4．输入 3 个数，求它们的和。

5．输入 2 个数，求它们的积。

新起点电脑教程

第 2 章

数据类型、运算符与表达式

本章要点

- 常用的数据类型
- 常量与变量的变义
- 标识符与关键字
- 运算符与表达式

本章主要内容

C 语言提供了丰富的数据类型，通过各种不同类型的数据常量和变量，用户可以灵活地处理各种问题。C 语言的标识符界定了用户可以使用及定义的字符集的范围，用户无论是使用系统给定的关键字，还是自定义标识符，都必须遵守标识符的使用规范。

C 语言也为用户提供了丰富的运算符，用户可以进行数学运算、关系运算、逻辑运算、位运算等多种复杂运算。灵活地使用运算符才能更好地使用 C 语言进行编程。

2.1 C 语言的数据类型

程序运行中处理的主要对象就是数据，解决不同的问题需要处理不同类型的数据，因此C语言中定义了多种不同的数据类型。在C语言中，数据类型可分为基本类型、构造类型、指针类型、空类型四大类。

↑扫码看视频

2.1.1 数据类型的分类

所谓数据类型，是按照被定义变量的性质、表示形式、占据存储空间的多少、构造特点等来划分的。在 C 语言中，数据类型可分为基本类型、构造类型、指针类型和空类型四大类。本节主要介绍基本类型及其基本运算。常用数据类型如图 2.1 所示。

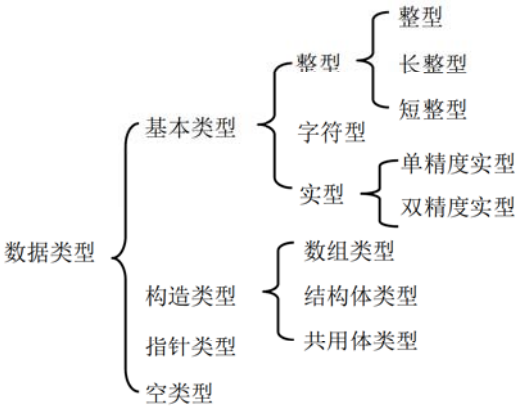


图 2.1 C 语言的数据类型

(1) 基本类型是 C 语言系统本身提供的，结构比较简单，其值不可以再分解为其他类型。

(2) 构造类型是由已定义的一个或多个基本类型构造而成。也就是说，一个构造类型的值可以分解成若干个“成员”或“元素”。每个“成员”都是一个基本数据类型或又是一个构造类型。在 C 语言中，构造类型有数组类型、结构体类型、共用体(联合)类型三种。

(3) 指针类型是一种重要的数据类型，其值用来表示某个变量在内存储器中的地址。可以表示复杂的数据结构，使用起来非常灵活，但是比较难理解和掌握。

(4) 在调用函数值时，通常应向调用者返回一个函数值。这个返回的函数值若具有一定的数据类型，应在函数定义及函数说明中给以说明。但是，也有一类函数调用后并不需

要向调用者返回函数值，这种函数可以定义为“空类型”，其类型说明符为 void。

2.1.2　数据类型的取值范围

不同的数据类型在内存中占用不同的存储空间，因此它们的取值范围也不同。如表 2.1 所示，为 C 语言中常用的基本数据类型所对应的字长(存储空间)和取值范围。

表 2.1　常用数据类型

类型标识符	名字	长度(字节)	取值范围
char	字符型	1	0～127
short int	短整型	2	-32768～32767
int	整型	2	-32768～32767
unsigned int	无符号整型	2	0～65535
long int	长整型	4	-2147483648～2147483647
float	单精度型	4	10^{-38}～10^{38}
double	双精度型	8	10^{-308}～10^{308}

注意：在 VC++6.0 环境中，int 型的字节长度和 long int 型的字节长度相同，都是 4 个字节。而在 TC 2.0 环境中，int 型的字节长度为 2 个字节。

2.2　常　　量

在 C 语言中，把程序运行过程中其值不能被改变的量称为常量。常量也被称为常数。常量可分为不同的类型，常用的有整型常量、实型常量、字符型常量、字符串型常量、符号型等等。

↑扫码看视频

2.2.1　整型常量

整型常量和实型常量也称为数值型常量。整型常量是由一个或多个数字组成，有正负值之分，但不能有小数点。整型常量有如下三种表示方法。

(1)　十进制整数：比如 257，458，-65，0。

(2)　八进制整数：在 C 语言中，用 0 开头的数来表示八进制数。例如 027 表示八进制数的$(27)_8$。

(3)　十六进制整数：在 C 语言中，用 0x 开头的数来表示十六进制数，例如 0xD4 表示

十六进制数的$(D4)_{16}$。

【例 2-1】屏幕输出三种整型常量。

程序代码

```
#include<stdio.h>
main()
{
    int x=1246, y=01246, z=0x1246;
    printf("%d, %d, %d\n", x, y, z);   /*  %d:以十进制格式输出*/
    printf("%o, %o, %o\n", x, y, z);   /*  %o:以八进制格式输出*/
    printf("%x, %x, %x\n", x, y, z);   /*  %x:以十六进制格式输出*/
}
```

运行结果如图 2.2 所示。

```
1246,678,4678
2336,1246,11106
4de,2a6,1246
```

图 2.2　十进制、八进制、十六进制整型常量

2.2.2　实型常量

在 C 语言中，带小数的数值被称为实数或浮点数，实型常量只使用十进制数。有以下两种表示形式。

(1) 十进制数形式：即数学中采用的实数形式，由正负号、整数部分、小数点、小数部分组成。

例如：7.68，−2.54，17.000，.123，123.，0.0。

(2) 指数形式：即数学中用到的指数形式，由正负号、整数部分、小数点、小数部分和字母 E(e)后面带正负号的整数组成。

例如：2700000 在数学中用指数可以表示为 2.7×10^6，C 语言中则表示为 2.7e6，0.00052 用指数形式可以表示为 5.2×10^{-4}，C 语言中则表示为 5.2E−4。

注意：① 字母 E(e)之前必须有数字。如 E5，E−7 是不合法的。

② 字母 E(e)之后的指数部分必须是整数。如 7e3.1 是不合法的。

③ 字母 E(e)与前后数字之间不得有空格。

2.2.3　字符型常量

字符型常量是由一对单引号括起来的单个字符。例如，‘A’，‘b’，‘，’，‘#’，‘9’等都是有效的字符型常量。

字符型常量的值是该字符集中对应的 ASCII 编码值(参见附录Ⅰ)。

注意：字符常量中的‘ 0’ ～‘ 9’与整型数据是不同的，例如‘ 9’对应的数值是 ASCII

值 57，而数值 9 对应的值是 9。

C 语言中还允许用一种特殊形式的字符常量，即以反斜杠字符‘\’开头的字符序列。前面使用过的 printf()函数中的‘\n’，代表一个“回车换行”符。这类字符称为“转义字符”，意思是将反斜杠‘\’后面的字符转换成另外的意义。常用的转义字符如表 2.2 所示。

表 2.2　常用的转义字符

转义字符	ASCII 码	字符	含义
\0	0	NULL	表示字符串结束
\n	10	NL(LF)	换行，将当前光标移到下一行的开头
\t	9	HT	水平制表
\v	11	VT	垂直制表
\b	8	BS	左退一格
\r	13	CR	回车，将当前光标移到本行的开头
\f	12	FF	换页
\′	39		单引号
\"	34		双引号
\\	92	\	反斜线
\ddd			1～3 位八进制数所代表的字符
\xhh			1～2 位十六进制数所代表的字符

2.2.4　字符串型常量

字符串型常量是由一对双引号括起来的字符序列，例如，“float”，“double”，“I ’m a Chinese”都是字符串型常量。

注意：“ A”和‘ A’是不同的，“ A”是字符串常量，字符 a 本身 1 个字节，加上系统自动加上的串尾标记‘ \0’，又占用 1 个字节，所以在内存占用 2 个字节长度。而‘ A’是字符常量，内存中只有存储字符 a 的 ASCII 码值，所以只占用 1 个字节长度。

2.2.5　符号型常量

C 语言中常用一个特定的符号来代替一个常量或一个较为复杂的字符串，这个符号称为符号常量。它通常由预处理命令#define 来定义。符号常量一般用大写字母表示，以便与其他标识符相区别。

预处理#define 又称为宏定义命令，一个#define 命令只能定义一个符号常量。因为它不是语句，所以结尾不用加分号。

【例 2-2】将π设为常量，给定半径，编程求圆的周长和面积。

程序代码：

```
#include  <stdio.h>
#define PI 3.14159
main( )
{
    float r,c,s;              /*定义圆的半径为 r，圆的周长为 c，圆的面积为 s*/
    scanf ("%f",&r);          /*键盘输入半径，放入变量 r 中*/
    c=2*PI*r;                                  /*利用圆周长公式求周长 c*/
    s=PI*r*r;                                  /*利用圆面积公式求面积 s*/
    printf("c=%.2f\n",c);        /*%.2f 表示把 c 的值按浮点型输出，保留两位小数*/
    printf("s=%.2f\n",s);        /*%.2f 表示把 c 的值按浮点型输出，保留两位小数*/
}
```

运行结果：

```
8↙
c=50.43
s=201.06
```

说明：用“#define　PI　3.14159“来定义 π 为常量，C 预处理程序在程序编译时，将该程序中所有的 PI 用 3.14159 来替换，但不作语法检查。

使用符号常量的优点如下。

(1) 增强程序可读性。在程序中定义一些具有一定意义的符号常量，能起到“见名知义”的作用。

(2) 简化输入程序。使用符号常量代替一个字符串，可以减轻程序中重复书写某些字符串的工作量。

(3) 增强程序的通用性和可维护性。如果一个程序中有多处使用同一个常量，这时，可把该常量定义为一个符号常量。若需要修改该常量，则只需要在定义处修改即可。可以做到一改全改，避免出现修改不完全或遗漏等错误。

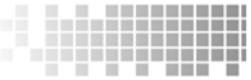
(4) 定义符号型常量，在编译时进行预处理，并不占用单独的内存空间。

2.3 变　　量

变量是指在程序运行过程中其值可以被改变的量。变量可以分为整型变量、实型变量、字符型变量、指针型变量等。

变量是 C 语言中重要的概念，学会正确定义变量是学好 C 语言的关键。

↑扫码看视频



2.3.1　变量的定义与初始化

程序中每一个用到的变量都应该有一个名字作为标识，称为“用户标识符”。变量名的命名规则应遵循标识符命名的规则。标识符由字母、数字或下划线组成，由字母或下划线开头，例如 x、y、a、b、x1、sum1、sum_t1 都是合法的变量名。

C 语言规定，变量必须先定义后使用。所谓的定义变量，实际上就是为其在内存中开辟一定数量的存储单元，而给变量赋值，则是将这个数值存储到该变量所代表的内存空间中。

定义不同类型的变量，在内存中占用不同的字节。例如：char 型变量分配 1 个字节，int 型变量分配 2 个字节，float 型变量分配 4 个字节。

对变量的定义通常放在函数的开头部分，变量只有从开始定义的位置才开始有实际意义。

变量定义格式为：

〈数据类型〉　〈变量名表〉；

例如：

```
int  x;          /*定义变量 x 为 int 型，系统给 x 分配 2 个字节的内存空间*/
x=1;             /*为变量 x 赋初值为 1 ，即把 1 存储到 x 所分配的内存空间中*/
int  x=1;        /*定义变量 x 的同时，给 x 赋初值 1*/
float  a,b;      /*定义变量 a，b 为 float 型，系统给 a，b 各分配 4 个字节的内存空间，a，
                   b 之间用，分开*/
a=0.04;
b=-4.56;
```

注意：C 语言的每个语句都以“ ；”号结束，因此句后的分号不能省略；同时定义两个以上变量时，中间以逗号分开。

2.3.2　整型变量

整型变量用来存放整型数据，即数学中的整数。整型变量有以下几种类型。

(1)　整型：用 int 表示(2 个字节)。

(2)　短整型：用 short int 或 short 表示(2 字节)。

(3)　长整型：用 long int 或 long 表示(4 字节)。

(4)　无符号整型：分为以下类型。

①　无符号整型：用 unsigned int 或 unsigned 表示(2 字节)。

②　无符号短整型：用 unsigned short int 或 unsigned short 表示(2 字节)。

③　无符号长整型：用 unsigned long int 或 unsigned long 表示(4 字节)。

无符号整型变量存储的是正整数，不能存放负数，因此存储单元中的全部二进制位都用来存放数据本身。而有符号整型则将首位用来存放负号。

短整型变量数值的表示范围是-32768～32767，无符号短整型数值的表示范围为 0～

65535，可以看出它们的取值范围是不同的。

【例 2-3】给定 x 值，通过函数 y=2x+1，计算 y 值。

程序代码：

```
#include "stdio.h"
main()
{
    int  x,y;
    scanf("%d",&x);
    y=2*x+1;
    printf("y=%d",y);
}
```

运行结果：

```
5↙
y=11
```

2.3.3 实型变量

实型变量又称为浮点型变量。按能够表示小数点后的精度，实型变量可分为三类。

(1) 单精度型：用 float 表示，在内存占用 4 个字节，有效数字 6～7 位。

(2) 双精度型：用 double 表示，在内存占用 8 个字节，有效数字 15～16 位。

(3) 长双精度型：用 long double 表示，在内存占用 16 个字节，有效数字 18～19 位。

单精度浮点型变量和双精度浮点型变量之间的差异体现在所能表示的数的精度上。一般单精度型数据占 4 个字节，有效位为 7 位，数值范围为 10^{-38}～10^{38}；双精度型数据占 8 个字节，有效位为 15～16 位，数值范围为 10^{-308}～10^{308}。

【例 2-4】单精度和双精度数据的输出比较。

程序代码：

```
#include <stdio.h>
main()
{
   float a;
   double b;
   a=123456.123456;
   b=123456.123456;
   printf("a=%f\nb=%lf\n",a,b);
}
```

运行结果如图 2.3 所示。

```
a=123456.125000
b=123456.123456
```

图 2.3 单精度和双精度型变量的区别

以上结果 a 为 float 型变量，只保证 7 位数据是有效的，后面的数据已经无效。而 b 为 double 型变量，可以保证 16 位数据的有效性。

2.3.4　字符型变量

一个字符型变量用来存放一个字符，在内存中占一个字节。将一个字符型常数赋值给一个字符型变量，并不是把该字符本身放到内存单元中去，而是将该字符对应的 ASCII 值(整数)存放到内存单元中。因此，字符型数据也可以像整型数据那样使用，用来表示一些特定范围内的整数并且进行计算。

【例 2-5】字符型数据与整型数据的转换。

程序代码：

```
#include <stdio.h>
main( )
{
  char c1,c2;                    /*定义 c1，c2 为字符型变量*/
  c1=97;                         /*将数值 32 赋给变量 c1*/
  c2='b';                        /*将字符 b 赋给变量 c2*/
  printf("%c,%c\n",c1,c2);       /*按字符格式输出 c1，c2 对应的字符*/
  printf("%d,%d\n",c1,c2);       /*按十进制输出 c1，c2 对应的 ASCII 码值*/
}
```

运行结果：

```
a，b
97，98
```

说明：可以从本例中看出，字符 a 的 ASCII 值是 97，字符 b 的 ASCII 值是 98，赋值的时候，无论是按字符型赋给变量一个字符值，还是按整型赋给变量一个整数值，都是将字符的 ASCII 值存入已开辟的内存单元中。输出的时候，按整型输出，则输出变量的 ASCII 值，如果按字符型输出，则输出变量的 ASCII 值所代表的字符。ASCII 值和字符可以很容易地互相转换。

【例 2-6】字符数据进行算术运算。

程序代码：

```
#include <stdio.h>
main( )
{
   char c1,c2;
   c1='A';
   c2='D';
   c1=c1+2;
   c2=c2-2;
   printf("%c,%c\n",c1,c2);        /*将 c1，c2 按所对应的字符格式输出*/
   printf("%d,%d\n",c1,c2);        /*将 c1，c2 按所对应的 ASCII 值输出*/
}
```

运行结果：

```
C，B
67，66
```

说明：A 的 ASCII 值为 65，A+2 的值为 67，如果按字符格式输出，结果为 C，按数值格式输出，结果为 67；D 的 ASCII 值为 68，D-2 值为 66，如果按字符格式输出，结果为 B，按数值格式输出，结果为 66。

2.3.5 定义不可变变量

C 语言提供了一个关键字 const 用来定义不可变变量，称为限定符或修饰符。例如：

```
const  float  PI=3.14;
     const  int  a=7;
```

const 可以在类型名前，也可以在类型名后，如果不写类型名，则默认为 int 型。因为定义的是不可变变量，在定义的时候必须赋初值。它们和正常变量一样，占有固定的内存空间，但内存空间的值是不可以改变的。它们和#define 定义的常量不同，常量不占用内存空间，没有类型问题，只是在编译的时候用字符简单代替而已。用 const 定义的变量严谨、明确，不会引起不必要的混乱，但是需要占用内存空间。

2.4 标识符与关键字

字符集是构成 C 语言的基本元素。用 C 语言编写程序时，所用的语句都是由字符集中的字符构成。C 语言的标识符都选自于 C 语言的字符集。

↑扫码看视频

2.4.1 标识符

C 语言中的标识符是用来标识变量名、常量名、函数名、数组名、类型名等程序对象的有效字符序列。C 语言对标识符有如下规定。

(1) 标识符只能由英文字母((A～Z，a～z)、数字(0～9)和下划线三种字符组成，且第一个字符必须为字母或下划线。

例如：a，x，x1，abc，memu_1，list_abc，al_2 等，都是合法的标识符。而 2a，x y，

a/b，x+y，a.b 等，则是不合法的标识符。

(2) 大小写字符代表不同的标识符。

例如：abc 与 ABC 是两个不同的标识符。一般变量名常用小写，符号常量名用大写。

(3) 不能使用 C 语言的关键字作为标识符。

(4) 对于标识符的长度，ANSI C 没有限制。但是，各个编译系统都有自己的规定和限制，Turbo C 2.0 限制为 8 个字符，超出的部分将被系统忽略。Visual C++ 6.0 基本没有限制，但是若标识符太长会影响输入速度。

2.4.2 关键字

C 语言规定的一些具有特定含义的、专门用来说明 C 语言的特定成分的标识符称为关键字。C 语言的关键字都是用小写字母来表示的。由于关键字具有特定的含义和用途，所以不能随便用于其他场合。否则，就会产生编译，虽然能通过，但是运行结果却错误的现象。以下列出常用的 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	sizeof	static
struct	switch	typedef	unsigned	union	void	volatile	while

2.4.3 预定义标识符与用户标识符

1. 预定义标识符

C 语言中还有一些预定义标识符也具有特殊的含义，尽量不要另作他用。比如编译预处理命令 define 等，或是库函数的名字 printf 等。从 C 语言的语法上，这些标识符可以另作他用，但是这将使这类标识符失去系统规定的原意，因此，为了避免误解，建议用户不要将其另作他用，以免带来不必要的麻烦。

2. 用户标识符

由用户根据需要自行定义的标识符称为用户标识符，一般用来给变量、常量、数组、函数或文件等命名。

用户标识符要遵循标识符的命名规则，尽量做到“见名知义”，选取具有正确含义的英文单词，增加程序的可读性。

用户标识符不能与关键字相同，如果相同，程序在编译时会给出出错信息。

用户标识符也尽量不要与预定义标识符相同，如果相同，程序不会报错，但是会使预定义标识符失去原定含义，也可能使程序出现错误的结果。

2.4.4 ASCII 码字符集

计算机中，所有的信息都用二进制代码表示。二进制编码的方式较多，应用最为广泛

的是 ASCII 码。我们使用的字符，在计算机中就是以 ASCII 码方式存储的。

ASCII 码是美国标准信息交换码(American Standard Code for Information Interchange)的缩写。它被国际标准化组织(ISO)认定为国际标准，详见附录 I。

ASCII 码分为标准 ASCII 码和扩展 ASCII 码。标准 ASCII 码在内存中占用一个字节，字节中的低 7 位用于编码，因此，可以表示 128 个符号。其控制符的编码值为 0～31，基本字符 0～9、A～Z、a～z 等编码值为 32～127(控制符用于计算机向外部设备输出一些特殊的命令，如控制打印机换行、换页等)。扩展 ASCII 码也称 8 位码，定义了 128～255 这 128 个数字所代表的字符。

2.5 运算符与表达式

C 语言的运算符非常丰富，本节主要介绍算术运算符、关系运算符、逻辑运算符等。C 语言的表达式是常量、变量、函数调用等用运算符连接起来的式子。凡是表达式都有一个值，即表达式的结果。

↑扫码看视频

2.5.1 C 语言的运算符

C 语言中一共有 44 个运算符，常用的运算符有以下几类。

(1) 算术运算符：+，－，*，/，%，++，－－。
(2) 关系运算符：<，<=，>，>=，==，!=。
(3) 逻辑运算符：&&，||，!。
(4) 赋值运算符：=，+=，－=，*=，/=，%=。
(5) 指针运算符：*，&。
(6) 条件运算符：?，：。
(7) 逗号运算符：，。
(8) 位运算符：&，|，～，^，<<，>>。
(9) 求字节运算符：sizeof。
(10) 特殊运算符：()，[ ]，->等。
(11) 分量运算符：．，-> 。
(12) 下标运算符：[ ]。
(13) 其他：如函数调用运算符等。

以上不同的运算符可以产生不同的表达式，这些表达式可以完成多种复杂的计算操作。运算符根据参与运算操作数的个数可分为：单目运算符、双目运算符、三目运算符。

例如，-5 中的负号，该运算符称为单目运算符；加、减、乘、除等运算符，称为双目运算符；条件运算符(?：)称为三目运算符。

2.5.2　运算符的优先级

由于 C 语言的运算符非常多，因此使用时变化也非常复杂，所以，C 语言规定了运算符的优先级和结合性。

当一个表达式中有多个运算符参加运算时，将按不同的先后次序进行运算。这种计算的先后次序称为运算符的优先级。

运算符的结合性，是指当一个操作数两侧的运算符具有相同优先级时，该操作数是先与左边还是先与右边的运算符相结合进行运算。从左向右的结合方向称为左结合性，从右向左的结合方向称为右结合性。

结合性是 C 语言的独有概念。除单目运算符、赋值运算符和条件运算符是右结合性外，其他运算符都是左结合性。 运算符的优先级与结合性如表 2.3 所示。

表 2.3　运算符的优先级和结合性

优先级	运算符	含　义	运算对象个数	结合方向
(高) 1	( ) [ ] -> .	括号 下标运算符 指向结构体成员运算符 结构体成员运算符		自左向右
2	! ~ ++ -- - (类型) * & sizeof	逻辑非运算符 按位取反运算符 自增运算符 自减运算符 负号运算符 类型转换运算符 指针运算符 取地址运算符 取长度运算符	单目运算符	自右向左
3	* / %	乘法运算符 除法运算符 求余运算符	双目运算符	自左向右
4	+ -	加法运算符 减法运算符	双目运算符	自左向右
5	<< >>	左移运算符 右移运算符	双目运算符	自左向右
6	<，<=，>，>=	关系运算符	双目运算符	自左向右

续表

优先级	运算符	含　义	运算对象个数	结合方向
7	== !=	等于运算符 不等于运算符	双目运算符	自左向右
8	&	按位与运算符	双目运算符	自左向右
9	^	按位异或运算符	双目运算符	自左向右
10	|	按位或运算符	双目运算符	自左向右
11	&&	逻辑与运算符	双目运算符	自左向右
12	||	逻辑或运算符	双目运算符	自左向右
13	? :	条件运算符	三目运算符	自右向左
14	=，+=，-= *=，/=，%= >>=，<<= &=，^=，|=	赋值运算符	双目运算符	自右向左
15(低)	,	逗号运算符		自左向右

2.5.3　算术运算与算术表达式

1．基本算术运算符

基本的算术运算符有 5 个，全部是双目运算符，分别是：

- +：加法运算符，如 13+6，15+x。
- -：减法运算符，如 a-b，36-7。
- *：乘法运算符，如 a*b，6*12。
- /：除法运算符，如 a/b，x/7。
- %：取余运算符(又称模运算)。如 a%b，6%2。

说明：(1) +、-、*/运算量可以是整数，也可以是实数。两个整数进行除法运算时，结果为整数，舍去小数部分，如 9/6 的，结果为 1。当参加运算的两个数中有一个为 float 型时，运算结果为 double 型，因为 C 语言对所有实数是按 double 型进行计算的，如 60.0/100=0.6。

(2) 取余(%)运算只能用于两个整型常量或整型变量，其运算结果为两整数整除后所得的余数。

(3) 当两个整数相除，除数或被除数有一个为负时，商为负；进行求余运算时，商的符号与被除数相同，如-5%3=-2，5%-3=2。

2．负号运算符

“-”也可用作单目运算符，称为负号运算符，如-5，-3.7。

3．自增(++)与自减(--)运算符

自增(++)与自减(--)运算符是 C 语言中两个最有特色的单目运算符。自增或自减运算的

作用是使变量的值增 1 或减 1，所以也称为增 1 或减 1 运算。如 i++相当于 i=i+1，i--相当于 i=i-1。

说明：(1)　自增运算符(++)与自减运算符(--)只能用于变量，不能用于常量或表达式。如 4++，(x+y)++都是不合法的。

(2)　++，--是单目运算符，结合方向自右向左，其优先级和负号运算符(-)一样。

例如：-a++。-和++是同一优先级，正常情况下，同一级别的运算符，运算时应从左向右运算，但是由于单目运算符的结合方向是自右向左，因此++先和运算量计算，所以上式应该等同于-(a++)。

(3)　++，--既可作为前置运算符，也可作为后置运算符，如 i++，i--，--i，++i 都是合法的表达式。无论作为前置运算符还是后置运算符，它们都有相同的作用，都是使变量加 1 或是减 1，但是作为表达式，却有不同的值。

例如：

```
int i=5;int x;
x=++i;       /*  i 的值增 1，为 6，表达式的值为 6，x=6*/
x= - - i;    /*  i 的值减 1，为 4，表达式的值为 4，x=4*/
x=i++;       /*  i 的值增 1，为 6，表达式的值为 5，x=5*/
x=i - - ;    /*  i 的值减 1，为 4，表达式的值为 5，x=5*/
```

【例 2-7】写出程序的运行结果。

程序代码：

```
#include  <stdio.h>
main( )
{
    int x=100;
    printf("x=%d\n",x++);   /* x++是后置运算，所以先输出 100 后加 1，x=101 */
    printf("x=%d\n",++x);   /* ++x 是前置运算，所以 x 先加 1 后输出，x=102  */
    printf("x=%d\n",x--);   /* x--是后置运算，所以先输出 102，后 x 减 1，x=101 */
    printf("x=%d\n",--x);   /* --x 是前置运算，所以 x 先减 1 后输出，x=100  */
}
```

运行结果：

```
x=100
x=102
x=102
x=100
```

【例 2-8】写出程序的运行结果。

程序代码：

```
#include  <stdio.h>
main()
{
    int i=3,x=0;              /*赋初值 i=3，x=0*/
```

```
    x=(++i)+(++i)+(++i);     /*因为++前置，先对 i 自加 3 次后 i=6,再 x=i+i+i=18*/
    printf("i=%d, x=%d\n",i,x);              /*打印 i=6,x=18*/
    i=3;x=0;                                 /*重新对变量赋初值 i=3，x=0*/
    x=(i++)+(i++)+(i++);
                /*++为后置，i=3 先求和运算，x= i+i+i =9， i 再自加 3 次后 i=6*/
    printf("i=%d, x=%d\n",i,x);  /*打印 i=6,x=9 */
}
```

运行结果：

```
i=6，x=16
i=6，x=9
```

4．算术表达式

用算术运算符和括号将运算对象如常量、变量和函数等连接起来的式子称为算术表达式。例如：a*b+c，x%2+y/2，4*sqrt(4c)，等，都是合法的算术表达式，其中 sqrt( )为开平方函数。

算术表达式书写规则如下。

(1) 所有字符必须写在同一水平线。

(2) 相乘的地方必须写上“*”符号，不能省略，也不能用“·”代替。

(3) 算术表达式中出现的括号一律用小括号，且一定要成对。

例如：求一元二次方程的根的公式：

$$x = \frac{-b \pm \sqrt{b^2 - 4ax}}{2a}$$

写成 C 语言的表达式如下：

```
x1=(-b+sqrt(b*b-4*a*c))/(2*a)
x2=(-b-sqrt(b*b-4*a*c))/(2*a)
```

5．强制类型转换

算术表达式中，双目运算符两边的操作数类型一致才能进行运算，所得结果的类型也与运算数的类型相同。不同类型的数据在进行混合运算时，必须先转换成同一类型，然后才能进行运算。不同类型的数据转换有两种方式：一种是自动类型转换，也称为隐式转换；另一种是强制类型转换，称为显式转换。

在整型、单精度型、双精度型数据之间进行混合运算时，将不同类型的数据由低向高转换成同一类型，然后进行运算。

例如：int a=7；float b=3.5；那么执行 a+b 时，按照 7.0+3.5 来计算。

【例 2-9】写出下面程序的运行结果。

程序代码：

```
#include  <stdio.h>
main( )
{
    int a,b;
```

```
    float c=45,d=7.18; /* 定义单精度型 c=45,实际上是 45.000000*/
    a=6.3;             /* 定义整型数 a=6.3，但实际上 a 只能取 6*/
    b=a+c+d;           /* a+c+d 的值是 58.45，但是由于 b 是整型变量，因此只能取 58*/
    printf("b=%d\n",b);
}
```

运行结果：

```
b=58
```

从上例可以看出，a，b，c，d 的数据类型不完全一样，因此，要根据定义的数据类型进行赋值，在计算时遵循由低到高转换的方法进行计算。这样虽然也能计算出一个结果，但是由于精度的丢失，结果的准确率大幅降低。因此，有时我们需要将操作数的类型强制转换为另一种类型进行计算。

转换格式如下：

```
(强制转换的类型名称) (操作数)
```

作用是把操作数强制转换为指定的类型。例如：(int)(a+b)，将 a+b 的结果强制转换成 int 型；又如：(float)a/b，将 a 强制转换成 float 型后，再进行除法运算，结果为 float 型。

【例 2-10】写出下面程序的运行结果。

程序代码：

```
#include <stdio.h>
main()
{
    float x=8.35;         /* 定义变量 x 为实型数据  */
    int a=5,b;
    b=( int)x %a;         /*实型数据不能取余运算，所以对变量 x 强制转换为 int 型 */
    printf("b=%d\n",b); /*7%5=2，所以打印 b=2 */
    printf("x=%x\n",x); /*x 的类型及值没有变，仍为 8.350000 */
}
```

运行结果：

```
b=2
x=8.350000
```

可以看到，变量经强制转换后，得到的是一个所需类型的中间变量，原来变量的类型并没有发生变化。

注意：自动(隐式)转换是计算机系统自动完成的转换，而强制转换是用户根据需要自己来进行的类型转换。

2.5.4　赋值运算符与赋值表达式

所谓赋值运算，是指将一个数据存储到某个变量对应的内存存储单元的过程。赋值运算符有两种类型：基本赋值运算符和复合赋值运算符。

1. 基本赋值运算符

C 语言的赋值运算符是“=”，它的作用是将赋值运算符右边表达式的值赋给其左边的变量。

例如：i=1，i=i+5 都是合法的赋值运算。

注意：如果“ =” 两侧的类型不一致，在赋值时需要进行自动转换。

2. 复合赋值运算符

C 语言允许在赋值运算符“=”之前加上其他运算符，构成其复合运算符。复合运算符多数为双目运算符。在 C 语言中，可以使用的复合赋值运算符有 10 个：+=、-=、*=、^=、%=、&=、^=、|=、<<=、>>=。

例如，a+=1，执行过程为：先对赋值运算符左右两侧进行运算，然后再把结果赋值给左边的变量。即相当于：a=a+1。

```
例如：x-=2;              /*等价于 x=x-2*/
      x/=5 ;             /*等价于 x=x/5*/
      x*=y+10;           /*等价于 x=x*(y+10) */
```

赋值运算符的结合方向是自右向左。C 语言采用复合赋值运算符，是为了使程序简练，提高编译效率。

注意：在书写复合赋值运算符时，两个运算符之间不能有空格，否则会出现语法错误。

3. 赋值表达式

由赋值运算符组成的表达式称为赋值表达式，例如 x=1。赋值表达式可以嵌套使用，例如 a=(b=4)，赋值表达式中的“表达式”，又是一个赋值表达式。

由于赋值运算符的结合方向是自右向左，因此，b=4 的括号可以不要，即 a=b=4，都是先求 b=4，然后载赋值给 a。

例如，a=10，表达式 a+=a-=a*a 的值为-180。因为，赋值运算符的结合方向是自右向左，所以运算顺序为：先进行计算 a-=a*a，即 a=a-a*a=10-10*10=-90。求出 a=-90 后，再计算 a=a+(a)=-90-90=-180。

4. 赋值语句

当在一个赋值表达式后面加上分号，就可以构成赋值语句。

```
例如：a=1        (赋值表达式)
      a=1;       (赋值语句)
      i++        (赋值表达式)
      i++;       (赋值语句)
```

2.5.5 逗号运算符与逗号表达式

逗号运算符为“，”。逗号表达式是用逗号运算符把两个表达式组合成的一个表达式。

格式：

```
〈表达式 1〉,〈表达式 2〉
```

说明：(1)　逗号表达式的执行过程是：先求“表达式 1”的值，再求“表达式 2”的值，“表达式 2”的值就是整个逗号表达式的值。例如：

```
a=8,a+10;
```

先对 a=8 进行赋值，然后计算 a+10，因此上述表达式执行完后，a 的值为 8，而整个表达式的值为 18。

(2)　一个逗号表达式可以与另一个表达式构成一个新的逗号表达式。例如：

```
(a=3*7,a*57),a+57;
```

构成一个逗号表达式，先计算 a=3*7 的值，a=21，然后计算 a*57 为值 1197，所以(a=3*7，a*57)的值为 1197。再计算第二个逗号表达式 a+57，此时 a 的值是 21，所以 a+57=78，那么逗号表达式“(a=3*7，a*57)，a+57”的值为 78。

(3)　逗号运算符是所有运算符中级别最低的。

【例 2-11】写出下面程序的运行结果。

程序代码：

```
#include <stdio.h>
main( )
{
    int x,y;
    x=7;
    y=(x-=10,x/5);    /* x 的初值为 7，减 10 后为-3，再除以 5，所以 y=0 */
    printf("y=%d\n",y);
}
```

运行结果：

```
y=0
```

2.6　思考与练习

本章详细介绍了 C 语言中常用的数据类型，C 语言中的标识符与关键字，运算符及优先级，C 语言的表达式，常量和变量的概念。C 语言丰富的运算符可以帮助读者轻松地解决各类问题，同时对于运算符的熟练运用，也是编写 C 语言程序的关键。

一、简答

1. C 语言中的数据类型主要有哪几类？
2. C 语言中有哪几种标识符？请举例说明。
3. 在 C 语言中，如何定义整型变量？如何定义实型变量？如何给变量赋初值？

4. 浮点变量有哪些表示形式？它们的数值范围都是什么？精度有什么区别？

5. 在 C 程序中，如何定义不可变的变量？它与一般变量有什么相同和不同之处？

6. 在 C 程序中，不可变变量与用 define 定义的常量在本质上有何不同之处？

7. C 语言中有哪些用于算术运算的运算符？请按它们的优先级别依次列出。

8. 下面的数据中哪些是 C 语言实型数的正确表示

(1)6.000　(2)5E+2　(3)0.3e1　(4).5　(5).e4　(6)1.234 e-4

(7)-0.0　(8)E-2　(9)6.　(10)6.0E+2.0　(11)7.e+0

9. 下列标识符中哪些可合法地用作用户标识符？

SADE　A#c　void　a*b　{g}　define　MAIN　b-a

ZZAP　i\xy　_6n　1456de　dele　$7　sin　int　s6a　_float

10. 下面哪些是不合法的常量？说明不合法的理由。

456，3.1415926，0662，‘ P’ ，‘ \n’ ，0xtty，0.897E-2，“ Good” ，2.6e-6.34

11. 下面对变量定义的语句哪些不正确？为什么？请改正。

(1)char c1，int a1；　(2)INT a，b；FLOAT x，y；　(3)a，b：char；

(4)char if；　(5)int x，y　(6)Int a：b：c；

(7)int a，b；float b，c；

12. 表达式 11/4 的结果是多少？表达式 11%4 的结果是多少？

13. 若有定义 int a=1,b=3；则表达式(a++)*(--b)执行后，a、b 及表达式的值各是多少？

14. 若有定义 int a=5,b；则执行 b=(a+=7,a/2)后，变量 a 和 b 的值分别为多少？

15. 请将下列代数式改写成 C 语言的表达式。

(1) -3ab+d-7ac　(2) 6x+8x-2x+89

(3) $\dfrac{5xy}{-x+\dfrac{1}{y+2(x+1)}}$　(4) $\dfrac{-(a-b)}{\dfrac{(a-b)-(a+b+1)}{2a}}$

16. 根据以下变量的定义，写出下列表达式的值。

int i=3，j=6，x=2，y=0；float m=2.7，n=5.0；

(1)x=j++　(2)(int)m+0.78*2　(3)n--　(4)x=(x=4,y=6,x+y)

(5)n+=-8　(6)m-=n+2　(7)m=--j　(8)m++,n+m,n++

(9)i/=-j-3　(10)i*=6,i+2*j　(11)++m　(12)y-=x+=(x=2,y=5)

(13)x=(j--)　(14)2%-4+-6%2　(15)j+(float)i+(int)(m++)

17. 下面的说法哪些是正确的？哪些是不正确的？

(1) 在 C 语言中，“ ；” 只是语句之间的分隔符，所以在每个句中都不能省略。

(2) 在 C 程序中，所用的变量都必须先定义后使用。

(3) 在 C 程序中，Printf 与 printf 是相同的，都表示 C 标准库提供的输出函数名字，scanf 和 Scanf 也是相同的，都表示 C 标准库提供的输入函数名。

(4) 程序中的变量 w 代表内存中的一个存储单元，所以经过 w=6 的运算后，也就是把 w 放入了这个存储单元中。

(5) 在 C 程序中，用 const 定义的变量，其值可以根据需要随时修改。

18. 若 i 为 int 类型，且有值 9，写出执行以下赋值运算后，i 和 k 中的值。

(1)k=++i　(2)k=--i　(3)k=i--　(4)k=i++

19. 若 x=8，y=4，经过以下运算，x 和 y 的值各是多少？

```
x=y;
y=x;
x=((x=y+7)-(y=x-3));
```

二、上机练习

1. 求以下程序的运行结果_____。

```
main()
{
    int i,j;
    i=10;
    printf("%d,%d", i++,i--);
}
```

2. 求以下程序的运行结果_____。

```
#include <stdio.h>
main()
{
    int a=35,n=7;
    a%=(n%=2);
    printf("a=%d\n", a);
}
```

3. 求以下程序的运行结果_____。

```
#include <stdio.h>
main()
{
    char a='W',b='e',c='l',d='l';
    a=a-16;
    b=b+10;
    c=c+3;
    d=d+2;
    printf("%c%c%c%c\n",a,b,c,d);
}
```

4. 求以下程序的运行结果_____。

```
#include <stdio.h>
main()
{
    int x;
    x=-7+2*75-9;
    printf("x1=%d\n",x);
    x=8+9%7-3;
```

```
    printf("x2=%d\n",x);
    x=-7*9%-3;
    printf("x3=%d\n",x);
}
```

5. 求以下程序的运行结果_____。

```
#include <stdio.h>
main()
{
    printf("%d,%c\n",57,57);
    printf("%d,%c,%o\n",47+10,47+10,47+10);
}
```

6. 求以下程序的运行结果_____。

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

7. 求以下程序的运行结果_____。

```
#include <stdio.h>
main()
{
    int a=5;
    a+=a-(a*=a);
    printf("a=%d\n",a);
}
```

8. 在以下程序中定义#define N 1000，求程序的运行结果_____。

```
#include <stdio.h>
#define N 1000
main()
{
    int a,b;
    a=N+67;
    b=N-46;
    printf("a=%d,b=%d\n",a,b);
}
```

三、编写程序

1. 已知边长 a 的值为 5，求立方体的体积。
2. 键盘输入三角形的两个内角角度，求另一个内角角度。
3. 已知梯形的上下边长及高，求梯形面积。