

第2章



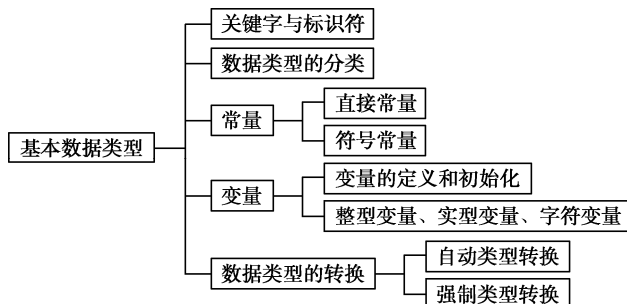
扫一扫下载本章教学课件

基本数据类型

教学导航

教	知识重点	1. 标识符命名规则; 2. 常量、变量的概念和用法
	知识难点	不同类型数据间的转换
	推荐教学方式	一体化教学: 边讲理论边进行上机操作练习, 及时将所学知识运用到实践中
	建议学时	4 学时
学	推荐学习方法	课前: 复习 C 语言程序的基本组成, 预习本章将学的知识 课中: 接受教师的理论知识讲授, 积极完成上机练习 课后: 巩固所学知识点, 完成作业, 加强对 Visual C++ 6.0 集成开发环境的使用
	必须掌握的理论知 识	1. C 语言中的关键字、标识符; 2. 常量的表示方法; 3. 变量的定义及初始化
	必须掌握的技能	熟练掌握 Visual C++ 6.0 集成开发环境的使用

知识分布网络





计算机的基本功能是处理数据，所以任何程序设计都要包括两个方面的工作。一方面是对数据进行描述，在程序中需要指定数据的类型及数据的组织形式，即数据结构；另一方面是对操作进行描述，也就是对数据的加工处理步骤与过程，即算法。

数据有类型之分，不同类型的数据具有不同的结构，在内存中的存放形式也不一样，因此编程时必须认真考虑如何选择和设计合适的数据结构。C 语言提供的基本数据类型比一般高级语言丰富。另外，用户还可以用基本数据类型组成一些复杂的数据类型，如数组、指针、结构体、联合体等。

2.1 关键字与标识符

2.1.1 关键字

关键字就是由 C 语言规定的具有特定意义、不能作为其他用途的字符串，通常也称为保留字。

C 语言的关键字分为两类。

(1) 类型说明符：用于定义和说明变量、函数或其他数据结构的类型。如第 1 章例题中用到的 `int`。

(2) 语句定义符：用于表示一个语句的功能，如 `for` 就是循环语句的语句定义符。

! C 语言的关键字由 ANSI 标准定义，共 32 个：

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

2.1.2 标识符

标识符是指以字母或下画线开头，后面跟字母、数字、下画线的任意字符序列。C 语言中的标识符可以分为系统预定义标识符和用户自定义标识符。

1. 系统预定义标识符

系统预定义标识符是由系统预先定义好的，每一个都有相对固定的含义，一般不做他用，以避免歧义，包括系统标准函数名和编译预处理命令等，如第 1 章例题中用到的 `main`、`printf` 和 `include` 等都是预定义标识符。

2. 用户自定义标识符

用户自定义标识符是用户根据编程需要自行定义的标识符，主要作为变量名、函数名、符号常量名、自定义类型名等。用户自定义标识符不能使用关键字，也尽量不要使用系统预定义标识符。用户自定义标识符必须满足以下规则：

- (1) 只能由字母 (`a~z`, `A~Z`)、数字 (`0~9`) 和下画线 (`_`) 组成。
- (2) 第一个字符必须为字母或下画线，不能是数字。
- (3) 自定义标识符不能和 C 语言的关键字相同。



(4) 同一字母的大小写被视为两个不同的字符，如 FUN 和 fun 是两个不同的标识符。

例如，smart、_score、sum1、name_1 都是合法的标识符；2age、x y、student-1、key.board、\$20.1 都是非法的标识符。

！在使用标识符时必须注意以下两点：

(1) 标准 C 语言不限制标识符的长度，但它受各种版本的 C 语言编译系统的限制，同时也受到具体机器的限制。例如，有些版本 C 语言中规定标识符前 8 位有效，当两个标识符前 8 位相同时，则被认为是同一个标识符；而有的 C 语言却允许长度达 32 个字符的标识符。

(2) 标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便阅读理解，做到“见名知意”。

2.2 数据类型的分类

在 C 语言源程序中对用到的所有数据都必须指定其数据类型。数据类型规定了该类型的数据在内存中的编码方式和长度、数据的取值范围、施加在该类型数据上的运算及运算结果的范围。

C 语言的数据类型比较丰富，其分类如图 2.1 所示。

由以上这些数据类型还可以构成更复杂的数据结构。例如，利用指针和结构体类型可以构成表、树、栈等复杂的数据结构。

本章只介绍 C 语言的基本数据类型，其他数据类型将在后面章节中陆续介绍。

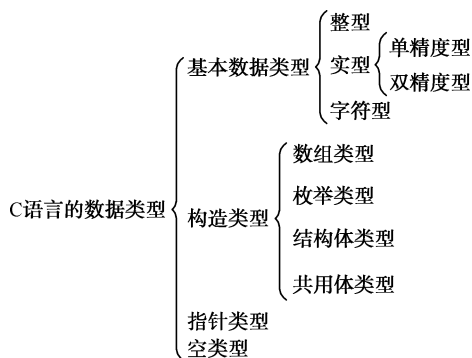


图 2.1 C 语言的数据类型

2.3 常量

在程序运行中，其值不能发生改变的量称为常量。常量分为直接常量和符号常量，现分别介绍如下。

2.3.1 直接常量

1. 整型常量

整型常量即整数，包括正整数、负整数和 0。在 C 语言中整型常量有 3 种表示形式。

(1) 十进制整数：有效数字为 0~9，如 20、123。

(2) 八进制整数：有效数字为 0~7，且以 0 开头，如 020 表示八进制数 20。

(3) 十六进制整数：有效数字为 0~9 和 a~f（或 A~F），且以 0x 开头，如 0x20 表示十六进制数 20，等于十进制数 32。



❗注意：在程序中，根据前缀来区分各种进制数，因此在书写常数时不要把前缀弄错以免结果不正确。

2. 实型常量

实型常量也称为实数或浮点数。在 C 语言中，实数只采用十进制。它有两种表现形式：十进制小数形式和指数形式。

(1) 十进制小数形式。由整数部分、小数点和小数部分组成，其格式为： $\pm$ 整数部分. 小数部分。

数字前的“+”或“-”表示数的正、负，“+”一般省略。当整数部分或小数部分为 0 时可以省略不写，但不能两者同时省略，而且小数点不能省略。例如，0.0，3.14，.25，0.13，5.0，300.等均为合法的实数。

(2) 指数形式。当数值特别大或特别小时，用指数形式表示更方便。指数形式由十进制数加字母“e”或“E”及指数部分组成。其一般形式为： aEn 。

其中， a 为十进制数， n 为十进制整数，其值为 $a \times 10^n$ 。

例如，1.2E5（等于 1.2×10^5 ），-3.8E2（等于 -3.8×10^2 ），0.5E-7（等于 0.5×10^{-7} ），-5.3E-2（等于 -5.3×10^{-2} ）。

❗注意：指数部分只能是整数，并且指数形式的 3 个组成部分都不能省略。例如，下面的表示方法是错误的，123，6.2E，e3，3e3.5。

3. 字符常量

字符常量是用一对单引号括起来的一个字符。其中单引号只起定界作用并不表示字符本身。

在 C 语言中，字符常量占一个字节存储空间，它存放的不是字符本身，而是字符的 ASCII 码值。例如，'!' 的 ASCII 码值为 33，'8' 的 ASCII 码值为 56，'A' 的 ASCII 码值为 65，'a' 的 ASCII 码值为 97。

❗注意：字符'8'和数字 8 的区别，前者是字符常量，后者是整型常量，它们的含义和在计算机中的存储方式都截然不同。

除了上面提到的单个普通字符，C 语言中还有一类叫转义字符。转义字符以反斜杠符（\）开头，后跟一个或几个字符。转义字符具有特定的含义，不同于字符原有的意义，故称“转义”字符。例如，在第 1 章例题中 printf 函数的格式串中用到的“\n”就是一个转义字符，其意义是“换行”。常用转义字符如表 2.1 所示。

表 2.1 常用转义字符

转义字符	意义	ASCII 码值（十进制）	转义字符	意义	ASCII 码值（十进制）
\a	响铃（BEL）	7	\\	反斜杠	92
\b	退格（BS）	8	\'	单引号字符	39
\f	换页（FF）	12	\"	双引号字符	34
\n	换行（LF）	10	\0	空字符（NULL）	0
\r	回车（CR）	13	\ddd	八进制表示任意字符	
\t	水平制表（HT）	9	\xhh	十六进制表示任意字符	
\v	垂直制表（VT）	11			



任意字符均可用反斜杠符 (\) 后跟该字符 ASCII 码值的八进制或十六进制 (以 x 作为前缀) 来表示。例如, 'a' 可表示为 '\101' 或 '\x41'; '\n' 可表示为 '\012' 或 '\x0A'。



使用转义字符时需注意:

- (1) 转义字符中只能使用小写字母, 每个转义字符只能看作一个字符。
- (2) '\v' 垂直制表符和 '\f' 换页符对屏幕没有任何影响, 但会影响打印机执行相应操作。
- (3) 使用不可打印字符时, 通常用转义字符表示。
- (4) 具有控制意义的转义字符, 在输入/输出时会引起设备完成相应的动作。

4. 字符串常量

字符串常量是由一对双引号括起来的零个或多个字符的序列。例如, "a"、"HELLO"、"1234"。

字符串常量在内存中存储时, 系统自动在字符串的末尾加一个字符串结束标志, 即 ASCII 码值为 0 的字符 NULL, 常用 '\0' 表示。因此, 在程序中长度为 n 个字符的字符串常量, 在内存中占有 $n+1$ 个字节的存储空间。

例如, 字符串 China 有 5 个字符, 作为字符串常量 "China" 存储于内存中时, 共占 6 个字节, 系统自动在后面加上 NULL 字符, 其存储形式为:

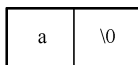
C	h	i	n	a	\0
---	---	---	---	---	----

字符串常量和字符常量是不同的常量, 它们之间主要有以下区别:

- (1) 字符常量由单引号括起来, 字符串常量由双引号括起来。
- (2) 字符常量只能是单个字符, 字符串常量则可以含零个、一个或多个字符。
- (3) 可以把一个字符常量赋予一个字符变量, 但不能把一个字符串常量赋予一个字符变量。在 C 语言中没有相应的字符串变量, 但是可以用一个字符数组来存放一个字符串常量, 在数组一章将予以介绍。
- (4) 字符常量占一个字节的内存空间。字符串常量占的内存字节数等于字符串中字符个数加 1。增加的一个字节中存放字符 '\0' (ASCII 码为 0), 这是字符串结束的标志。例如, 字符常量 'a' 和字符串常量 "a" 虽然都只有一个字符, 但在内存中的情况是不同的, 如图 2.2 所示。



(a) 'a' 的存储表示



(b) "a" 的存储表示

图 2.2 'a' 和 "a" 的存储表示

2.3.2 符号常量

在 C 语言中, 允许用一个标识符来代表一个常量, 即常量可以用“符号”来代替, 代替常量的符号就称为“符号常量”。符号常量一般由大写英文字母表示, 以区别于一般用小写字母表示的变量。符号常量在使用前必须先定义, 定义的形式是:

```
#define 符号常量名 常量
```

例如:

```
#define PI 3.14
#define PRICE 30
```



```
#define YES 1
#define NO 0
```

这里定义 PI、PRICE、YES、NO 为符号常量，其值分别为 3.14、30、1、0。#define 是 C 语言的预处理命令，它表示在对程序进行编译时，首先将出现符号常量的地方用对应的常量替换。

❗ 使用符号常量时，应注意以下几项：

- (1) 符号常量不同于变量，它的值在其作用域内不能改变，也不能再被赋值。
- (2) 符号常量定义式不是 C 语句，行尾没分号。
- (3) 使用符号常量的好处有两个。一是含义清楚，如前面定义的符号常量 PI，代表圆周率；二是在需要改变一个常量值时能做到“一改全改”。

❓ 读一读 2.1 字符常量和整型常量的算术运算。

```
#include<stdio.h>
main()
{
    printf("%d\n",65+'B');
}
```

运行结果为：131

因为字符常量 'B' 在内存中以其 ASCII 码的形式 (66) 存储，所以 65+'B' 实际上是执行 65+66=131，因此运行结果为输出整数 131。

❓ 读一读 2.2 转义字符的使用。

```
main()
{
    printf(" ab c\t dc\r f\n");
    printf("f\t b\b k");
}
```

运行结果为：

```
fab_ _c_ _dc
f_ _ _ _k
```

分析：此例中，用 printf() 函数直接输出双引号中的字符，第 1 个输出语句在第 1 行第 1 列开始输出 “_ab_ _c_”；然后遇到 “\t”，它的作用是跳到下一个制表位，一个制表位为 8 个字符，所以在第 9 列上输出 “dc”；当遇到 “\r” 时，回车（不换行）在第 1 行第 1 列输出 f；当遇到 “\n” 时，回车换行到第 2 行第 1 列。第 2 个输出语句在第 2 行第 1 列先输出 f，然后跳到第 9 列准备输出，又遇到连续两个控制符 “\b”，则向后退 2 格在第 7 列输出 k。

❓ 练一练 2.1 分析如下程序的运行结果。

```
#include <stdio.h>
main()
{
```



```
printf("\101 \x42 C\n");
printf("I say:\"How are you?\"\n");
printf("\\C Program\\n");
printf("Turbo \'C\'");
}
```

练习指导：注意单引号、双引号及反斜杠的表示。

2.4 变量

在程序运行过程中，其值可以发生改变的量称为变量。变量具有 3 个基本要素：名字、类型和值。

(1) 变量名：每个变量都必须有一个名字。变量的命名应遵循标识符的命名规则。

(2) 变量类型：变量类型是变量所能存储数据的类型。

(3) 变量值：变量在程序运行过程中，占据一定的内存存储单元，用来存放变量的值。不同类型的变量所占用的内存单元的大小不同，变量值随着变量赋值而改变。在程序运行过程中，通过变量名来引用变量的值。

2.4.1 变量的定义

在 C 语言中，变量必须在使用（对变量进行操作）之前定义。定义时确定了变量的名字和数据类型。其格式如下：

类型说明符 变量名列表；

其中，类型说明符是任意有效数据类型的说明符，规定了变量所能存储数据的类型（通常也称为变量类型）。变量名列表包括一个或多个标识符，每个标识符之间用逗号“,” 隔开。例如：

```
int a,b,c;
float data;
```

第 1 行定义了数据类型均为整型，变量名分别为 a、b 和 c 的 3 个整型变量；第 2 行定义了一个浮点型变量 data。

2.4.2 变量的初始化

在定义变量的同时用赋值运算符“=”给变量赋初值，称为变量的初始化，它是在系统编译时完成的。一旦变量被初始化后，它将保留此值直到被改变为止。例如：

```
int a=4; /*指定 a 为整型变量，初值为 4*/
float f=4.56; /*指定 f 为实型变量，初值为 4.56*/
char c='a'; /*指定 c 为字符变量，初值为'a'*/
```

也可以对被定义变量的一部分赋初值，如：int a=1,b=-3,c; 表示 a、b、c 为整型变量，只对 a、b 初始化，a 的值为 1，b 的值为-3。

2.4.3 整型变量

整型变量可用来存储整型数据。C 语言中用 int 表示基本整型变量，此外还可以在 int 前加上两类修饰符构成多种类型的整型变量：按数据占内存空间的大小可分为短的（short）和



长的 (long); 按数据的正、负可分为有符号的 (signed) 和无符号的 (unsigned), 通常有符号类型前的 signed 可以省略。

(1) 基本整型以 int 表示; 短整型以 short int 或 short 表示; 长整型以 long int 或 long 表示。

(2) 无符号整型以 unsigned int 表示; 无符号短整型以 unsigned short int 或 unsigned short 表示; 无符号长整型以 unsigned long int 或 unsigned long 表示。

在 VC 编译系统中, 整型变量所占的字节数和取值范围见表 2.2。

表 2.2 各种整型变量占用的字节数和取值范围

类 型 名	占用字节数	取 值 范 围
int	4	-2 147 483 648~2 147 483 647
short[int]	2	-32 768~32 767
long[int]	4	-2 147 483 648~2 147 483 647
unsigned[int]	4	0~4 294 967 295
unsigned short[int]	2	0~65 535
unsigned long[int]	4	0~4 294 967 295

❗注意: 在实际应用过程中, 应根据数据的特性及取值范围来选择整型变量的类型。

例如, 用 age 表示平均年龄变量, 取值范围为 0~100; 用 x 表示硬盘的字节数, 取值范围为 $0 \sim 2^{16}$, 则这两个变量定义为:

```
int age;                /*将 age 定义为整型变量*/
unsigned long int x;     /*将 x 定义为无符号长整型变量*/
```

2.4.4 实型变量

实型变量分为单精度型和双精度型。单精度型变量定义时使用关键字 float, 双精度型变量定义时使用关键字 double。例如:

```
float x,y;              /* 指定变量 x、y 为单精度型变量 */
double z;               /* 指定变量 z 为双精度型变量 */
```

两种类型之间的差异仅仅体现在所能表示的数的精度上。在 C 语言中实型变量所占字节数、有效数字位数和取值范围如表 2.3 所示。

表 2.3 实型变量

类 型	占用字节数	有 效 数 字	取 值 范 围
float	4	6~7	绝对值: $3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$
double	8	15~16	绝对值: $1.7 \times 10^{-308} \sim 1.7 \times 10^{308}$

❗注意: float 型数据的有效位数是 6~7 位, 在进行赋值和计算时会产生误差。

**实例 2.1 实型变量赋值示例。**

```
#include<stdio.h>
main()
{
    float x=12345678.9;
    double y=12345678.9;
    printf("\nx=%f,y=%f",x,y);
}
```

运行结果为: x=12345679.000000, y=12345678.900000

从运行结果可以看出, 由于 float 型变量只能接收 7 位有效数字, 因此从第 8 位开始数据不准确。而变量 y 为 double 型, 则能全部接收上述 9 位数字并存储在变量 y 中。

2.4.5 字符变量

字符变量用来存放字符常量, 每个字符变量被分配一个字节的内存空间, 因此只能存放一个字符。

字符变量的定义及赋值举例如下:

```
char c1, c2;    /*定义 c1 和 c2 为字符变量*/
c1='a';        /* 对 c1 赋值*/
c2='b';        /* 对 c2 赋值*/
```

! 注意: 字符是以 ASCII 码值的形式存放在字符变量的内存单元之中的。

字符 a 的十进制 ASCII 码值是 97, b 的十进制 ASCII 码值是 98, 上面的字符变量 c1、c2 分别被赋予了'a'和'b', 实际上是在 c1、c2 两个单元内分别存放 97 和 98 的二进制代码, 即

```
c1: 0110 0001
c2: 0110 0010
```

所以, 也可以把字符变量看成是整型变量。C 语言允许对整型变量赋予字符变量, 也允许对字符变量赋予整型变量。对整型变量赋予字符变量时, 实际上是把该字符的 ASCII 码值赋给整型变量; 对字符变量赋予整型变量时, 实际上是把 ASCII 码值与该整型变量的低字节相等的字符赋给字符变量。同理, 如果把字符变量按整型变量输出, 则输出的是字符的 ASCII 码值; 如果把整型变量按字符变量输出, 则输出的是 ASCII 码值与整型变量低字节相等的字符。

? 读一读 2.3 字符变量示例。

```
#include<stdio.h>
main()
{
    char a, b;
    a=65;                                /*65 是 A 的 ASCII 码值, 所以此句等于 a = 'A';*/
    b=66;                                /*66 是 B 的 ASCII 码值, 所以此句等于 a = 'B';*/
    printf("%c,%c\n ", a, b);            /*%c 表示将变量 a 和 b 按字符型输出*/
    printf("%d,%d\n", a, b);            /*%d 表示将变量 a 和 b 按整型输出*/
}
```

程序运行后显示的结果为:



A, B
65, 66

因为字符变量可以看成整型变量, 所以字符变量可以参与数值运算, 即用字符的 ASCII 码值参与运算。

❓ 练一练 2.2 分析如下程序的运行结果。

```
#include<stdio.h>
main()
{
    char c1, c2;
    c1='a';
    c2='b';
    c1=c1-32;
    c2=c2-32;
    printf("%c %c", c1, c2);
}
```

练习指导: (1) 字符变量参与了数值运算; (2) 小写字母的 ASCII 码比其大写字母的 ASCII 码大 32。

2.5 数据类型的转换

在 C 语言中, 整型、实型和字符型数据间可以混合运算。在运算过程中, 如果一个运算符两侧参与运算的数据类型不同, 则系统按“先转换, 后运算”的原则, 首先将数据自动转换成同一类型, 然后在同一类型的数据之间再进行运算。数据的转换方式有两种, 即自动转换和强制转换。

2.5.1 自动转换

自动转换的规则有两点: 一是低类型数据必须转换成高类型数据; 二是赋值符号“=”右边的数据类型转换成左边的数据类型。具体转换规则如图 2.3 所示。

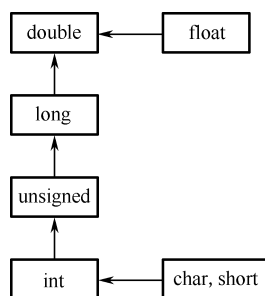


图 2.3 数据类型转换规则

❗ 数据类型自动转换时需要注意以下几点:

(1) 图 2.3 中横向的箭头表示必定发生的转换。如字符数据 char 和短整型数据 short 必须先转换成整型数据 int; 实型数据 float 必须先转换成双精度型数据 double, 然后再进行各种运算。

(2) 图 2.3 中纵向箭头表示当运算对象为不同数据类型时的转换方向。例如, 某个 int 型数据和某个 long 型数据运算时, 应先将 int 型转换成 long 型, 然后再进行运算, 其结果是 long 型。

(3) 当进行赋值运算时, 将赋值符号右边的类型转换成赋值符号左边的类型, 其结果为赋值符号左边的类型。如果赋值符号右边为实型 float, 左边为整型 int, 则转换时应去掉小数部分; 如果右边是双精度型 double, 左边是实型 float, 则转换时应做四舍五入处理。



2.5.2 强制转换

上面运算中数据类型的转换都是系统自动进行的,但有时设计者需要自己实现数据类型的转换,这种转换形式称为强制转换。其一般形式如下:

(数据类型名)(表达式)

其中,数据类型名表示待强制转换的类型;表达式表示强制转换的对象。

例如:

```
(float)a      /*把变量 a 转换为实型*/  
(int)(x+y)    /*把 x+y 的结果转换为整型*/
```

! 注意: 无论是强制转换还是自动转换,都只是为了本次运算的需要而对变量的数据长度进行的临时性转换,并不改变原变量的类型。

? 读一读 2.4 分析以下程序运行结果。

```
#include<stdio.h>  
main()  
{  
    float x;  
    int i;  
    x=3.6;  
    i=(int)x; /*强制转换*/  
    printf("x=%f,i=%d",x,i);  
}
```

运行结果为: x=3.600000,i=3

! 注意: 将较高类型强制转换为较低类型时可能发生精度损失。



扫一扫看
数据类型
转换分析
视频

练一练 2.3 分析如下程序的功能。

```
#include<stdio.h>  
main()  
{  
    float pi=3.14159;  
    int s,r=5;  
    s=r*r*pi;  
    printf("s=%d\n",s);  
}
```

练习指导: 将 $r*r*pi$ 赋值给 s 时,存在类型转换。

2.6 基本数据类型常见错误及解决方法

在数据类型常量、变量的定义和使用过程中,初学者易犯一些错误。下面列出与本章内



容相关的一些程序错误，分析了其原因，并给出了错误现象及解决方法。

示例 1 `int a=1;`

`temp=a;`

错误现象：系统报错——`'temp':undeclared identifier`。

错误原因：变量未定义就使用。

解决方法：增加变量 `temp` 的定义，再使用该变量。

示例 2 `int temp;`

`tep=1;`

错误现象：系统报错——`'tep': undeclared identifier`。

错误原因：变量名拼写错误。

解决方法：查看对应的变量及其定义，保证前后一致。

示例 3 `int temp;`

`Temp=1;`

错误现象：系统报错——`"Temp":undeclared identifier`。

错误原因：未区分大小写字母。

解决方法：查看对应的变量及其定义，区别大小写字母。

示例 4 `printf("Hello");`

`int x=0;`

错误现象：系统报错——`missing ';' before 'type'`。

错误原因：变量定义位置在其他可执行语句后面。

解决方法：将变量集中在语句块开始处定义，变量定义不能放在可执行语句中间。

示例 5 `int a;`

`printf("%d",a);`

错误现象：系统警告——`local variable 'a' used without having been initialized`。

错误原因：使用了未赋值的变量，其值不可预测。

解决方法：养成对变量初始化的习惯，保证访问前有确定值。

示例 6 `int a=b=1;`

错误现象：系统报错——`'b': undeclared identifier`。

错误原因：在定义变量时，对多个变量进行初始化。

解决方法：改为 `int a=1,b=1;`

示例 7 `#define PI 3.14;`

`double s,r=1.0;`

`s=PI*r*r;`

错误现象：系统报错——`illegal indirection, '*': operator has no effect` 等语法错误。

错误原因：定义符号常量时后面加分号。

解决方法：`#define` 是编译预处理命令，不是 C 语言语句，后面不要加分号。

示例 8 `int a=10000;`

`a=a*a*a;`

`printf( "%d" ,a);`



C 语言程序设计案例教程

错误现象：系统无报错或警告，但是输出结果不正确。

错误原因：未考虑数值溢出的可能。

解决方法：预先估计运算结果可能的范围，采用取值范围更大的类型，如 `double`。

示例 9 `int a,b`

`a=1;b=2;`

错误现象：系统报错——missing ';' before identifier 'a'。

错误原因：语句之后丢失分号。

解决方法：找到出错位置，添加分号。

知识梳理与总结

本章首先介绍了 C 语言数据类型的分类，然后结合实例介绍了常量与变量的定义及用法，最后介绍了整型数据、实型数据和字符数据之间的相互转换。现将本章需要掌握的知识点总结如下。

(1) 标识符是指常量、变量、语句标号及用户自定义数据类型、函数的名称。在 C 语言中命名标识符时，务必遵循一定原则，避免不合法标识符的使用。

(2) C 语言的数据类型有：基本类型、构造类型、指针类型和空类型。其中，基本类型包括整型、实型（包括单精度实型和双精度实型）和字符型。

(3) 常量是指在程序运行过程中其值不能被改变的量。常量分为直接常量和符号常量，直接常量通常有 4 种类型：整型常量、实型常量、字符型常量、字符串常量。整型常量有十进制、八进制和十六进制表示。

(4) 在程序运行时，其值能被改变的量叫变量，变量必须先定义后使用。变量的类型由定义语句中的数据类型标识符指定。系统根据变量类型分配相应的存储空间，存放变量的值。通过变量初始化可以给变量赋初值。不能直接使用未经赋值的变量，因为它的值是一个不确定的数据。

(5) 在进行混合运算时，如果一个运算符两侧的运算对象的数据类型不同，则系统按“先转换，后运算”的原则，首先将数据自动转换成同一类型，然后在同一类型的数据间进行运算。

自测题 2



扫一扫看
本自测题
答案

一、选择题

1. 以下不能定义为用户标识符的是 ()。
A. `scanf2` B. `Void` C. `_3com _` D. `int`
2. 以下选项中合法的用户标识符是 ()。
A. `short` B. `_2Test` C. `3Dmax` D. `A.dat`
3. 下列数据中，数值或字符常量不正确的是 ()。
A. `0.825e2` B. `5L` C. `0xabcd` D. `o13`
4. 以下字符中，不正确的 C 语言转义字符是 ()。



- A. '\'
B. '018'
C. '\xaa'
D. '\t'

5. 已知大写字母 A 的 ASCII 码值是 65, 小写字母 a 的 ASCII 码是 97, 则用八进制表示的字符常量 '\101' 是 ()。

- A. 字符 A
B. 字符 a
C. 字符 e
D. 非法的常量

6. 以下选项中可作为 C 语言合法整数的是 ()。

- A. 10110B
B. 0386
C. 0Xffa
D. x2a2

7. 以下符合 C 语言语法的实型常量是 ()。

- A. 1.2E0.5
B. 3.14.159E
C. .5E-3
D. E15

8. 已定义 ch 为字符型变量, 以下赋值中错误的是 ()。

- A. ch='\';
B. ch=62+3;
C. ch=NULL;
D. ch='\xaa';

9. 已定义 c 为字符型变量, 则下列赋值中正确的是 ()。

- A. c='97';
B. c="97";
C. c=97;
D. c="a";

10. C 语言中最基本的数据类型包括 ()。

- A. 整型、实型、逻辑型
B. 整型、实型、字符型
C. 整型、字符型、逻辑型
D. 整型、实型、逻辑型、字符型

11. 在 C 语言中, 合法的字符常量是 ()。

- A. '\084'
B. '\x43'
C. 'ab'
D. "\0"

12. 已知在 ASCII 代码中, 字母 A 的序号为 65, 以下程序的输出结果是 ()。

```
#include <stdio.h>
main()
{char c1='A ', c2='Y '; printf("%d,%d\n", c1, c2); }
```

- A. 输出格式非法, 输出错误信息
B. 65, 90
C. A, Y
D. 65, 89

二、填空题

1. C 语言中, 直接常量包括_____、_____、_____和_____。
2. C 语言中, 数据类型包括_____、_____、_____和_____。
3. C 语言中的变量实际代表内存中的一个_____。
4. 在 C 语言中, 整数可用_____进制数、_____进制数与_____进制数 3 种来表示。

三、简答题

1. C 语言的基本数据类型有几种, 分别是什么? 指出各种数据类型的关键字。
2. 常量和变量的区别是什么?
3. 在程序中如何使用变量?
4. 字符和字符串表示形式有什么区别? 'a'和"a"有什么不同? 9 和'9'有何不同? 空字符'\0'和空格字符' '有何区别?
5. 对 C 语言来说, 下列标识符中哪些是合法的, 哪些是不合法的?
total _debug Large&Tall Counter1 begin_ SUM 5S max ¥123
NO! do for_c l anguage printf (xyz) a com PI



上机训练题 2

写出下列程序的运行结果

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

运行结果为: _____

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

运行结果为: _____

```
3. #include <stdio.h>
main()
{
    float a;
    a=3.1415927;
    printf("a=%f",a);
}
```

运行结果为: _____

```
4. #include <stdio.h>
main()
{
    float f=5.75;
    printf("(int)f=%d,f=%f\n",
        (int)f,f);
}
```

运行结果为: _____



扫一扫看
本训练题
答案