

二级C 精品课程2-3：基本数据类型 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/181/2021_2022__E4_BA_8C_E7_BA_A7C___E7_c97_181150.htm

在C中，数据具有不同的类型，类型定义了变量可存储的数值范围以及可进行的操作。变量是用于内存中保存数据的，每个变量都必须有确定的数据类型，C语言的数据类型如图2-1所示。在图2-1中，把数据类型划分为基本的数据类型和复合的数据类型，我们也可把数据类型分为内置的类型和用户定义的类型两大类，用户定义的类型在使用以前，必须先定义，包括：结构、类、枚举和联合类型；内置的类型是指直接被C提供的类型，也就是说，是除用户定义的类型以外的其它类型。图2-1 C数据类型

从语法上来说，void类型也是基本的类型，但是，它不是一个完整的类型，只能作为更复杂类型的一部分。没有void类型的变量，它或者用于指定一个函数（在第五章介绍）没有返回值，或者作为指针（在第七章介绍）类型，表示该指针指向未知类型的变量，例如：`void x. //错误，没有void变量`
`void f(). //函数f不返回值`
`void *pv. //指针pv指向未知类型的变量`

布尔型、字符型、整型和浮点型也称为算术类型；枚举、指针、数组、引用、结构、联合和类称之为复合数据类型，它们能通过其它数据类型进行构造。在基本的数据类型char、int前可以添加修饰符，以改变基本类型的意义，可用的修饰符有long、short、signed和unsigned四种，另外，双精度型前可以加long修饰符。基本的数据类型及其表示范围，可参见表2-3。

类型名	类型字节表示范围
char	字符型 1-128 ~ 127
unsigned char	无符号字符型 0-255

~255 signed char 有符号字符型 (与字符型相同) 1-128 ~127 int 整型*与机器有关 unsigned int 无符号整型*与机器有关 signed int 有符号整型 (与整型相同) *与机器有关 short int 短整型 2-32,768 ~ 32,767 unsigned short int 无符号短整型 20~65,535 signed short int 有符号短整型 (与短整型相同) 2-32,768 ~ 32,767 long int 长整型 4-2,147,483,648 ~2,147,483,647 signed long int 有符号长整型 (与长整型相同) 4-2,147,483,648 ~ 2,147,483,647 unsigned long int 无符号长整型 40~4,294,967,295 float 浮点型 43.4E /- 38 (7位有效数字) double 双精度型 81.7E /- 308 (15位有效数字) long double 长双精度型 101.2E /- 4932 (19位有效数字) unsigned 和 signed 只用于修饰 char 和 int, 且 signed 修饰词可以省略。当用 unsigned 修饰词时, 后面的类型说明符可以省略。例如: signed int n. // 与 "int n." 等价 signed char ch. // 与 "char ch." 等价 unsigned int n. // 与 "unsigned n." 等价 unsigned char ch. // 与 "unsigned ch." 等价 short 只用于修饰 int, 且用 short 修饰时, int 可以省略, 即: short int n. // 与 "short n." 等价 long 只能修饰 int 和 double。当用 long 修饰 int 时, int 可以省略, 即: long int n. // 与 "long n." 等价 int 和 unsigned int 类型占用一个机器一个字 (word) 的字节。在16位操作系统上, 它们占用2个字节; 在32位操作系统上, 它们占用4个字节。各种数据类型精度由低到高排列如图2-2: 图2-2 如果一个双目运算符两边的操作数类型不同, 先要将它们转换为相同的类型, 即较低类型转换为较高类型, 然后再参加运算。所谓类型的高低, 跟所占的存储空间大小有直接关系, 所占存储空间越大的类型, 级别越高。图中横向的箭头表示必须的转换, 如两个 float 型数参加运算, 虽

然它们类型相同，但仍要先转换成double型再进行运算，结果亦为double型。纵向箭头表示当运算符两边的操作数为不同类型时的转换，如一个long型数据与一个int型数据一起运算，需要先将int型数据转换为long型，然后两者再进行运算，结果为long型。所有这些转换都是由系统自动进行的，使用时只需了解结果的类型即可。程序2-2

```
#include <string.h>
void
main(void){ char a = 'x'; int b = 3; float f = 2.5678; double d = 5.2345; long e = 32L; cout << d / c * e - b << endl; }
```

下面我们来分析一下这段程序：（1）进行d / c运算时，要将c转换成double型，运算的中间结果为double型；（2）进行e * f运算时，将f转换为long型，运算的中间结果为long型；（3）进行a - b运算时，将a转换为int型（数值为ASCII码值120），运算的中间结果为int型；（4）当（3）的中间结果与（1）的中间结果运算时，将（3）的中间结果转换为double型，运算的中间结果为double型；（5）当（4）的中间结果与（2）的中间结果运算时，将（2）的中间结果转换为double型，得出最后结果。于是，程序最后的运行结果为55.038515。如果数据是从较高类型转换成较低类型，这意味着类型的精度或表示范围降低，可能造成数据丢失。下面两个表分别列出了有符号数之间和无符号数之间的转换：

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com