



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

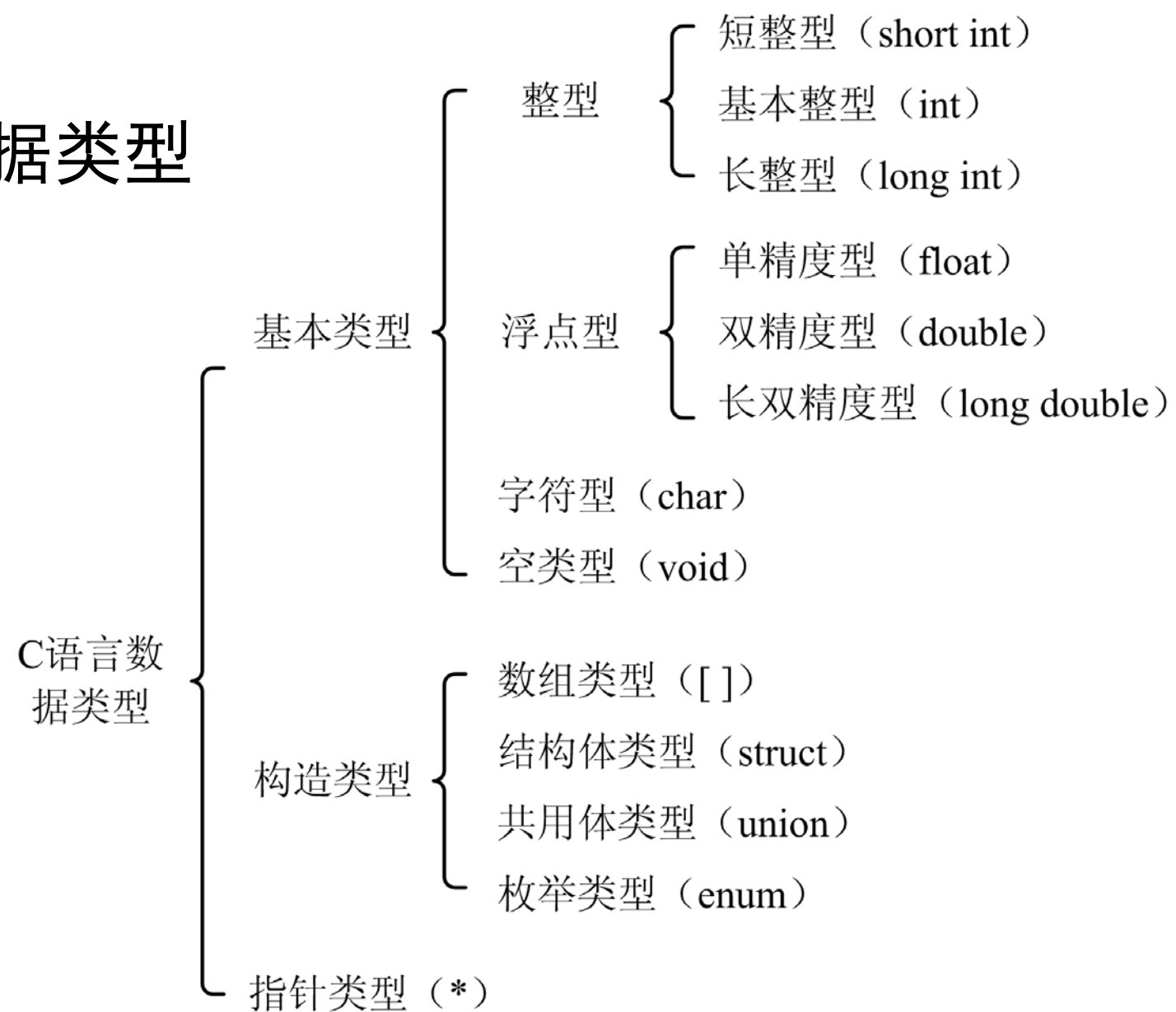
主讲：姜学锋，计算机学院

将简单数据引入C程序中

- ◆ 1、数据类型
- ◆ 2、数值常量

2.1 数据类型

► C语言内置数据类型



2.1 数据类型

表2-1 基本类型数据的内存长度和数值范围

类型	类型标识符	内存长度（字节）	数值范围	精度
整型	[signed] int	4	-2147483648～+2147483647	
无符号整型	unsigned [int]	4	0～4294967295	
短整型	[signed] short [int]	2	-32768～+32767	
无符号短整型	unsigned short [int]	2	0～65535	
长整型	[signed] long [int]	4	-2147483648～+2147483647	
无符号长整型	unsigned long [int]	4	0～4294967295	
字符型	[signed] char	1	-128～+127	

2.1 数据类型

续表2-1 基本类型数据的内存长度和数值范围

类型	类型标识符	内存长度（字节）	数值范围	精度
无符号字符型	unsigned char	1	0~255	
单精度型	float	4	$3.4\times10^{-38} \sim 3.4\times10^{38}$	7
双精度型	double	8	$1.7\times10^{-308} \sim 1.7\times10^{308}$	16
长双精度型	long double	同上/12	同上/ $1.2\times10^{-4932} \sim 1.2\times10^{4932}$	同上 /19

2.1.1 整型

- ▶ C语言整型分为长整型（long int）、基本整型（int）和短整型（short int）
- ▶ long int可以简写为long、short int可以简写为short。
- ▶ int型数据的内存长度与系统平台相关，通常int型为机器的一个字长，short型不比int型长，long型不比int短。

2.1.1 整型

- ▶ 短整型数123在内存中的存储形式为

0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

S

- ▶ 短整型数-123在内存中的存储形式为

1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

S

2.1.1 整型

- ▶ 整型还分有符号（signed）和无符号（unsigned）类型，其中signed书写时可以省略。

有符号

0000000000000000	0111111111111111	1000000000000000	1111111111111111
0	32767	- 32768	- 1

无符号

0000000000000000	1111111111111111
0	65535

2.1.1 整型

- 一般地，超过最大值的有符号整型数值会向上溢出变成负数，超过最小值的数据会向下溢出变成正数。

图2.2 short型的溢出

01111111111111111111

+ 000000000000000001

10000000000000000000

32767

+

1

- 32768 (补码)

向上溢出

10000000000000000000

+ 11111111111111111111

10111111111111111111

- 32768

-

1

32767

向下溢出

2.1.2 浮点型

- ▶ C语言浮点型又称实型，分为单精度（float）、双精度（double）和长双精度（long double）3种。
- ▶ 在VC中规定float型在内存中占用4个字节，提供7位有效数字；double型和long double型在内存中占用8个字节，提供16位有效数字。在GCC中long double型在内存中占用12个字节，提供19位有效数字。

2.1.2 浮点型

- ▶ 因为浮点型数据长度和精度是有限的，所以浮点数存在舍入误差和计算误差。
- ▶ 一个较大的浮点数与一个很小的浮点数做加法时，由于精度限制使得很小的浮点数被忽略了，从而使得这样的加法无意义。
- ▶ 两个浮点数做比较，由于有误差很难做到绝对相等，只能通过它们差的绝对值小于一个很小的数来判断是否近似相等。

2.1.2 浮点型



【例2.1】

浮点型数据的误差。

2.1.2 浮点型

例2.1

```
1 #include <stdio.h>
2 int main()
3 {
4     float  a=0.00000678f, b=0.00000123f;
5     double c=0.00000678 , d=0.00000123;
6     double e=1000000000000000000000000.0;
7     a=a+111111.111f; //精度范围外的大浮点数与小浮点数相加
8     b=b+111111.111f; //精度范围外的大浮点数与小浮点数相加
9     c=c+111111.111;  //精度范围内的大浮点数与小浮点数相加
10    d=d+111111.111;  //精度范围内的大浮点数与小浮点数相加
11    e=e+111111.111f; //精度范围外的大浮点数与小浮点数相加
12    printf("a=%.16f , b=%.16f\n",a,b); //输出单精度浮点型a,b
13    printf("c=%.16lf , d=%.16lf\n",c,d); //输出双精度浮点型c,d
14    printf("e=%lf\n",e); //输出双精度浮点型e
15    return 0;
```

2.1.2 浮点型

例2.1

程序运行屏幕

```
1 #include <stdio.h>
2 int main()
3 {
```



```
a=111111.1093750000000000,
b=111111.1093750000000000
c=111111.1110067800000000,
d=111111.1110012300100000
e=100000000000000000110000.000000
```

5. 范围内的大浮点数与小浮点数相加
6. 范围外的大浮点数与小浮点数相加

```
printf("%16f\n", a, b); //输出单精度浮点型a,b
printf("%16lf\n", c, d); //输出双精度浮点型c,d
//输出双精度浮点型e
```

2.1.3 字符型

- ▶ C语言字符型分为有符号（signed char）和无符号（unsigned char）两种，其中signed书写时可以省略。
- ▶ 字符型数据在内存中占用1个字节，采用二进制形式存储。

2.1.3 字符型

- ▶ 字符型数据可以存储整型数值，有时也称为字节型。字符型数据存储整数时的内存形式与整型一样，只不过其数值范围要小得多。
- ▶ 字符'A'在内存中的存储形式为

0	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

'A'的ASCII码值

2.1.3 字符型



【例2.2】

字符型数据与整型数据的赋值与运算。

2.1.3 字符型

例2.2

```
1 #include <stdio.h>
2 int main()
3 {
4     int i, j;
5     char c1, c2;
6     c1='a'; //字符数据赋值给字符型
7     c2=98; //整数数据赋值给字符型
8     i='A'; //字符数据赋值给整型
9     j=66; //整数数据赋值给整型
10    printf("i=%d , j=%d , c1=%c , c2=%c\n", i, j, c1, c2);
11    printf("c1-32=%c\n", c1-32); //字符型可以进行减法运算
12    return 0;
13 }
```

2.1.3 字符型

例2.2

程序运行屏幕

```
1 #include <stdio.h>
2 int main()
3 {
```



```
i=65, j=66, c1=a, c2=b
c1-32=A
```

```
c1=%c , c2=%c\n", i, j, c1, c2);
-32); //字符型可以进行减法运算
```

2.2 常量

- ▶ 常量（constant）是指程序中其值不能被修改的数据，分为字面常量和符号常量。
- ▶ 从字面形式即可识别的常量称为字面常量（literal constant），例如64、3.1415926和'A'等。

2.2.1 整型常量

- ▶ 一个整型常量可以用3种不同的方式表示：
 - ▶ (1) 十进制整数。
 - 以非零十进制数1~9组成的整数，例如13579,-24680等。
 - ▶ (2) 八进制整数。
 - 以0开头和八进制数0~7组成的整数，例如0, 012, 0177等。
 - **012+12=?**
 - **029+29=?**

2.2.1 整型常量

- ▶ (3) 十六进制整数。
 - 以0x或0X开头和十六进制数0~9、A~F或a~f组成的整数（字母大小写任意），例如0x1234，0xab，0xCF等。

2.2.1 整型常量

► 整型常量从字面上区分数据类型的方法是：

- （1）整型常量默认的类型为int型。根据系统平台，若int和long型数据占用内存大小相同，则一个int型常量也是long型常量。若int和short型数据占用内存大小相同，则一个int型常量也是short型常量。
- （2）一个整数如果其值在范围内，则它是short型。

2.2.1 整型常量

- (3) 一个整数如果其值超出上述范围，但在范围内，则它是long型。
- (4) 在一个整数值后面加一个字母l或L，则它是long型。例如123是int型，123L是long型。后缀符号一般用L而不用小写的l，避免与数字1混淆。
- (5) 整数默认是signed型，在一个整数后面加一个字母u或U，则它是unsigned型。

2.2.1 整型常量

► 示例

```
18      //十进制表示
022     //八进制表示
0x12    //十六进制表示

0        //signed int
168      //signed int
168U     //unsigned int
2147483647 //signed long
-1L      //signed long
65535Lu  //unsigned long
```

2.2.2 浮点型常量

► 一个浮点型常量可以用两种不同的方式表示：

- (1) 小数形式。由小数点、十进制整数和小数组成的浮点数，如 1.234、-567.89 等。整数和小数可以省略其中之一，但不能省略小数点，如 .123、123.、0.0 等。
- (2) 指数形式，又称科学记数法表示。以 $f E n$ 或 $f e n$ 格式组成的浮点数，其中 E 或 e 表示以 10 为底的幂， n 为指数且必须是整型， f 可以是整数或小数。
- $1.2e3 = 1.2 \times 10^3$

2.2.2 浮点型常量

- ▶ 浮点型常量默认为double型。若在浮点数后面加一个字母f或F，则它是float型。若在浮点数后面加一个字母l或L，则它是long double型。
- ▶ 示例

```
f=1.23;    //默认double  
f=1.23f;   //float  
f=1.23L;   //long double
```

CP 程序设计