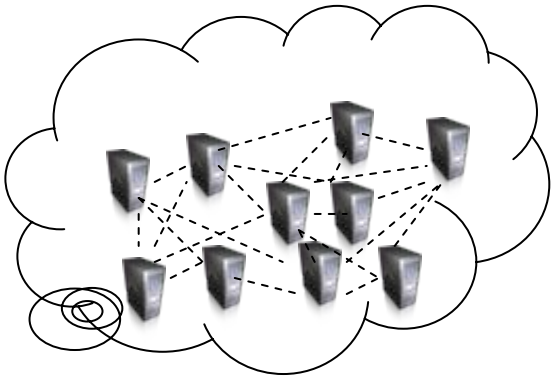
厦门大学非计算机专业本科生公共课

（2011-2012第2学期）



C语言程序设计

林子雨

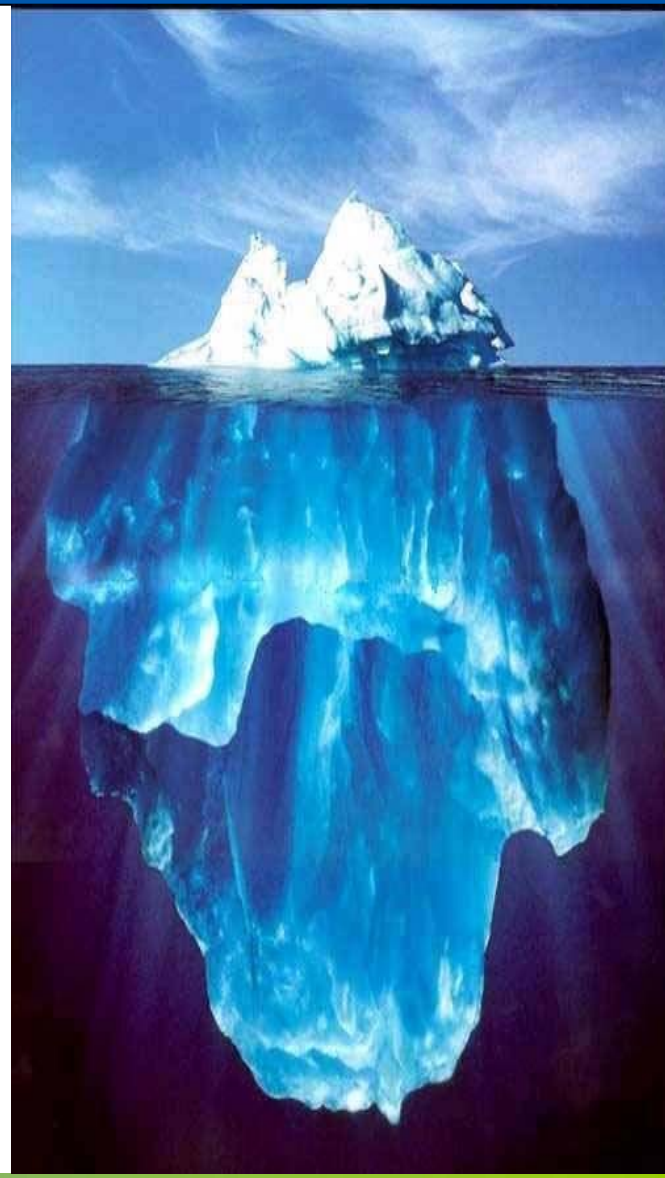
厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn

个人主页：**http://www.cs.xmu.edu.cn/linziyu**

课程提要

- 第一章 绪论
- 第二章 C语言基础
- 第三章 结构化程序设计
- 第四章 选择结构
- 第五章 循环结构程序设计
- 第六章 函数
- 第七章 编译预处理
- 第八章 数组
- 第九章 结构体、共用体和枚举类型
- 第十章 指针



第2章 C语言基础

2.1 C语言的词汇与词法

2.2 C语言的数据类型

2.3 常量与变量

2.4 运算符和表达式

2.1 C语言的词汇与词法

2.1.1 基本字符集

2.1.2 关键字

2.1.3 特定字

2.1.4 标识符

2.1.5 运算符

2.1.6 分隔符

2.1.7 字面常量

2.1.1 基本字符集

- 大小写英文字母、数字、特殊符号 + - * / % > < = ! , .?:;^~’ ” ( ) [ ]
- { } $ |# \ & 空格字符 换行字符
- C语言中字母的大小写是有区别的
  - –两个词 if和IF是不同的

2.1.2 关键字

关键字，也称保留字，是C语言中具有特定作用和含义的单词，在程序中不能另作其他用途。

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

2.1.3 特定字

特定字是一些用在C语言的预处理命令和库函数名中的单词，这些字都是由编译系统规定的，有特定含义。

- 预处理命令
  - define include ifdef endif
- 函数名
  - scanf printf
  - main

2.1.4 标识符

标识符：以字母或下划线打头，由字母、数字字符和下划线组成的字符序列

- 作用
  - 变量名，常量名，函数名，类型名
- 词法
  - 只能由字母、数字、下划线组成
  - 必须由字母或下划线打头
- 几点说明
  - 不能是关键字或者特定字
  - 大小写有别
  - 标识符的起名风格

2.1.5 运算符

运算符也称操作符，告诉计算机如何操作数据：

- (1) 算术运算符： +、-、*、/、%、++、--
- (2) 关系运算符： >、<、>=、<=、 = =、 !=
- (3) 逻辑运算符： ！ 、&& 、||
- (4) 位运算符： << 、>> 、~ 、| 、^ 、&
- (5) 赋值运算符： =、+=、-=、*=、/=、%=、&=、|=、^=、<<=、>>=
- (6) 条件运算符： ? :
- (7) 逗号运算符： ，for(i=1,s=0 ;i<10; i++) s=s+i;
- (8) 指针运算符： *、 &
- (9) 求字节数运算符： sizeof
- (10) 分量运算符： . 、->
- (11) 下标运算符： [ ]

2.1.6 分隔符

- 分隔符用来界定或分割语句中的语法成分（像文章中的标点符号）
  - 分号; 表示一个语句的结束 （编译预处理命令和{}后面不能加分号）
  - 空格 逗号,在两个相邻的保留字或标识符之间起分割作用。连续多个空格和单个空格的作用相同
    - 如 int a 和 int a, b
  - 单引号‘字符的开始和结束
  - 双引号“字符串的开始和结束
  - 花括号{}函数体的开始和结束 复合语句的开始和结束
  - /* */注释的开始和结束
  - 运算符也能分割单词
    - 如a=3 和 a = 3

2.1.7 字面常量

- **字面常量：**在程序中直接写出值的常量

1. 整型常量

- 十进制 八进制十六进制
- 如何判断一个整型常量的进制：十进制不能以0开头，由0~9数字组成，八进制由0开头，由0~7数字组成，十六进制由0x或0X开头，由0~9 A~F(a~f)组成
- 不同进制的转换（65，0101，0x41）
- 不同进制输出：%d, %o, %x（参见p34例2.1.1）

2. 浮点型常量

1. 只采用十进制，有小数点表示法和指数表示法两种形式
2. 小数点表示法，可以省略小数点前后的0。如 28.56, -0.37, 129. , .23
3. 指数表示法，代表e(E)前的数字乘以10的指数次幂。e(E)前面不能没有数字，e(E)后面必须是整数，且不能使用()。如0.314e2, 9.2e-3等是合法的，e5、4e3.1、2E(-6)等不合法。

习题p63二.15

2.1.7 字面常量

3、字符型常量：

- 用单引号括起来的单个字符，可以参与数值运算
  - 如’a’（ASCII码值97）, ’Y’, ’ ’, ‘A’(65),’?’,’B’(66)
- 转义序列表示法
  - ASCII码值小于32的特殊字符，如响铃、回车、换行、退格（8），无法直接用字符常量的上述表示方法，需要用转义序列表示法
  - 采用’\ddd’或’\xhh’表示字符常量
    - 如：’\102’和’\x42’都表示’B’,
- 转义字符,所谓转义字符则是以“\”开头的特殊字符。常用的转义字符为:\n（换行，十进制10，八进制012),值得注意的是，一个转义字符仅仅算一个字符。

2.1.7 字面常量

字符编码（ASCII码）

L \ H	0000	0001	0010	0011	0100	0101	0110	0111
0000	NUL	DLE	SP	0	@	P	‘	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	“	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	,	7	G	W	g	w
1000	BS	CAN	)	8	H	X	h	x
1001	HT	EM	(	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	’	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

‘a’的ASCII码值（二进制）：01100001 ，（十进制）97

换行符ASCII码值（二进制）：00001010 ，（十进制）10，（八进制）12

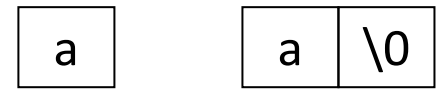
2.1.7 字面常量

字符型常量：转义字符

转义字符	转义字符的意义	ASCII代码
\n	回车换行	10
\t	横向跳到下一制表位置	9
\b	退格	8
\r	回车	13
\f	走纸换页	12
\\	反斜线符"\"	92
\'	单引号符	39
\”	双引号符（仅在字符串中才要反斜杠）	34
\a	鸣铃	7
\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	

2.1.7 字面常量

- 4、字符串常量
  - 由双引号括起来的一个或多个字符,也可以没有字符.
    - 例：”How do you do!”
    - 双引号 “Please enter \” Yes or \”No:”
  - 存储,每个字符占一个字节,另外每个字符串末尾加一个’\0’ 作为结束标志
  - 说明：字符串和字符
    - ”a”与’a’不要混淆, ” a”为字符串常量,内存占两个字节



字符’a’　字符串”a”

2.1.7 字面常量

举例

```
#include <stdio.h>
void main()
{ printf("%c--%d,%c--%d\n",'a','a','A','A');
  printf("%d--%c,%d--%c\n",'a'-32,'a'-
   32,'A'+32,'A'+32);
}
```

运行结果为
a--97,A--65
65--A,97--a

2.2 C语言的数据类型

- 2.2.1 C语言数据类型概述
- 2.2.2 C语言的基本数据类型
- 2.2.3 举例

2.2.1 C语言数据类型概述

- 数据类型
  - 基本类型
    - 整型
      - 短整型（short）
      - 整 型（int）
      - 长整型（long）
      - 无符号整型(unsigned int)
    - 浮点型（实型）
      - 单精度型（float）
      - 双精度型（double）
    - 字符类型（char）
    - 空值型(void)
  - 构造类型
    - 数组类型
    - 结构体类型（struct）
    - 共用体类型（union）
    - 指针类型

2.2.2 C语言的基本数据类型

整型：用于描述整数,在C 语言中,整型用int 来说明,分为基本型、短整型、长整型、无符号型。

类型	类型标识符	在内存中所占字节数	数值范围
基本型：	int	4	-2147483648~2147483647
短整型：	short int	2	-32768~32767
长整型：	long int	4	-2147483648~2147483647
无符号整型：	unsigned int	4	0~4294967295
无符号短整型：	unsigned short	2	0~65535
无符号长整型：	unsigned long	4	0~4294967295

2.2.2 C语言的基本数据类型

浮点型

单精度（float ）　4字节，范围 $\pm(3.4\times10^{-38}\sim3.4\times10^{38})$

有效精度：6~7位

双精度（double）　8字节，范围 $\pm(1.7\times10^{-308}\sim1.7\times10^{308})$

有效精度：15~16位

更高精度（long double）16字节，$\pm(1.1\times10^{-4932}\sim1.1\times10^{4932})$

有效精度：18~19位

浮点型存储数据有误差，例如：

```
#include <stdio.h>
        void main()
        { float a=12.3,b;
          double c;
          b=c=12345.678;
          printf("a=%f,b=%f,c=%1f\n",a,b,c);
        }
```

运行结果：

```
a=12.300000,b=12345.677734,c=12345.678000
```

2.2.2 C语言的基本数据类型

字符型

- 字符数据在内存中并不是存储字符本身，而是存储字符所对应的ASCII值。其存储形式和整型数是类似的。也正是这个原因使C中字符型数据和整型数据可以混合使用。这样的字符存储模式，有利于我们对字符数据进行处理，我们可以用熟悉的数字运算对ASCII码进行处理而最终得到字符的运算结果。

例 字符型变量与整数的关系举例。

```
#include <stdio.h>
main()
{ char ch;
  int i;
  ch='A';
  ch=ch+32;
  i=ch;
  printf("%d is %c\n",ch, i );
  printf("%c is %d\n",ch,ch);
}
```

运行结果：
97 is a
a is 97

(1) 字符型变量的值可以赋给一个整型变量；
(2) 字符型变量可以和整数混合运算；
(3) 整数值可以以字符的形式输出,反之亦然。

2.2.2 C语言的基本数据类型

空值型

- 用关键字void表示, void型的值集为空集,可出现在函数定义的头部,表示该函数没有返回值.
- 例如

```
void main()
{
    …
}
```

2.2.3 举例

例2.2.1 观察运行下面程序后sum和ave的值

```
#include <stdio.h>
main( )
{
int a, b, c,sum;  //定义变量a、b、c、sum为整型
float ave;        //定义变量ave为浮点型
a=3; b=6; c=2;    //分别给整型变量a、b、c赋值3、6、2
sum=a+b+c;        //将变量a的值3、b的值6和c的值2之和11赋给整型变量sum
ave=sum/3;        //将变量sum的值11除以3的商3赋给浮点型变量ave
printf("%d, %f\n" , sum, ave); //按指定的格式输出sum和ave的值
}
```

运行结果：
sum=11,ave=3.000000

2.3 常量与变量

- 2.3.1 常量
- 2.3.2 变量
- 2.3.3 举例

2.3.1 常量

常量是在程序运行过程中其值不能被改变的量

常量
- 字面(直接)常量
  - 整型常量
    - 十进制数： 以非0开始的数，如：220、-560、+369等。
    - 八进制数： 以0开始的数，如：06、0106、0677等。
    - 十六进制数：以0X或0x开始的数：0X123、0xab、 0X4e
  - 浮点型常量(十进制实数)
    - 一般形式：由数字、小数点以及必要时的正负号组成。如：29.56、-56.33、0.056、.056、0.0等。
    - 指数形式：相当于科学计数法，将形如$a\times10^b$的数值表示成如下形式： aEb 或 aeb
      如： 2.956E3、-0.789e8、.792e-6等
  - 字符常量：用两个英文单引号限定的一个字符，如：'a' 、'9' 、'Z'等。占一个字节的存储空间，存放的是其ASCII码值(一个整数)
    另外，还有以“\”开头的转义字符常数，如'\n'。
  - 字符串常量：用英文双引号限定的一个字符序列，如："Hello world" 、"a"
    占用一段连续存储单元，每个字符占一个字节。尾部加字符串结束标志符'\0' ，如字符串"STUDY"的存储形式 ：
- 符号常量

定义格式： const 类型名 常量名=值

#define <常量名> <值常量>

2.3.1 常量

例2.3.1 编写程序，计算并输出半径为10的球表面积和球体的体积。

```
#include <stdio.h>
#define R  10          //定义符号常量R
#define PI 3.14159     //定义符号常量PI
main( )
{       float s,v;
                s= 4*PI*R*R;
                v=s*R/3;
                printf("s=%f  v=%f\n",s,v);
}
```

运行结果：

```
s=1256.637085   v=4188.790039
```

使用符号常量的好处：

①含义清楚。

②在需要改变一个常量时能做到“一改全改”。

2.3.2 变量

- 2.3.2.1 变量名
- 2.3.2.2 变量的类型
- 2.3.2.3 变量的值
- 2.3.2.4 变量的地址
- 2.3.2.5 变量的定义
- 2.3.2.6 变量的初始化

2.3.2.1 变量名

- 变量名由英文字母、数字0-9、下划线_组成。英文字母区分大小写，不得使用空格和其他特殊字符，如?和*等
- 变量名的第一个字符不能是数字
- 变量名不能是C语言中的关键字和特定字
- 变量名可以为任意长度，但最好不要超过31个字符
- 变量名应使读者容易明白其含义

2.3.2.2 变量的类型

- 每个变量都存储特定类型的值，因此，每个变量都有确定的值
- 变量的类型决定了该变量能取何种值、对其能进行何种运算以及所需要的存储空间大小

2.3.2.3 变量的值

- 变量的值是指变量所表示的数据
- 它是与这个变量相关的存储单元的内容
- 在程序运行过程中，变量的值可以改变

2.3.2.4 变量的地址

- 在程序运行时，程序中的每个变量都将会拥有一块内存空间，从而它们都有一个内存地址
- 变量的地址，为存储该变量值的内存空间的首地址
- 在高级语言程序设计中，通常不必关心变量的地址

2.3.2.5 变量的定义

语法格式：

数据类型 变量名1,变量名2,…,变量名n;

或

数据类型 变量名1=初值1,变量名2=初值2,…,变量名n=初值n;

```
int x=1,y=x-1;
float z=1.0;
char c='A';
double w;
```

注意:

(1)定义一个变量只是在它的名称和它的值类型之间建立联系。若在定义它时未初始化，此时该变量的值一般是不确定的，不能用来计算。因此，在使用变量之前，一定要给该变量**赋值**。

(2)在C程序中使用变量之前，必须对使用的变量进行**声明**，使得编译器能对变量的操作进行类型检查。

声明

（1）定义性声明：比如变量定义，int a=0;

（2）非定义性声明：extern 数据类型名 变量名；（extern int a;）

```
extern int a=5;//错误，非定义性声明不能对变量初始化
```

2.3.2.6 变量的初始化

数据类型　变量名=表达式；

例如：

int a=6,b=8;

int a,b,c=5;

int a=3,b=3,c=3;

不能写成：int a=b=c=3；

2.3.3 举例

例2.3.3整型数据的溢出。

```
#include  <stdio.h>
main( )
{  short int a,b;                    //定义短整型变量a和b
   long int c;                       //定义长整型变量c
   a=32767;
   b=a+10;
   c=32768;
   printf("a=%d,b=%d,c=%ld\n",a,b,c);
}
```

运行结果：
a=32767,b=-32759,c=32768

2.3.3 举例

例2.3.4浮点型数据的存储误差。将一个有效数字超过7位的实数赋给浮点型变量，然后输出该浮点型变量。

```
#include<stdio.h>
main()
{
float a=12.3, b;
double c;
b=12345.678; c=12345.678;
printf("a=%f,b=%f,c=%lf\n",a,b,c);}
```

运行结果：
a=12.300000,b=12345.677734,c=12345.678000

2.4 运算符和表达式

- 2.4.1 运算符和表达式概述
- 2.4.2 算术运算符和算术表达式
- 2.4.3 类型转换
- 2.4.4 赋值运算和赋值表达式
- 2.4.5 逗号运算符和逗号表达式

2.4.1 运算符和表达式概述

- 运算符一般要携带若干运算对象，运算对象称为操作数。根据操作数的个数，运算符可以分为单目运算符、双目运算符和三目运算符。
- 表达式由运算符和相应的操作数以及用于描述运算先后次序的括号构成。
- 注意：单独的常量、变量和函数调用也是表达式。
- 为了表达式的书写方便（可以少写一些括号），C引进了运算符优先级和结合方向的概念。
- 运算符的结合方向（结合性）规定了同优先级运算符相遇时候的运算次序。
- C运算符有两种结合方向：自左向右结合（左结合性），自右向左结合（右结合性）。

2.4.2 算术运算符和算术表达式

- 单目算术运算符：-（取负），+（取正）
- 双目算术运算符：+、-、*、/
- 变量的自增++、自减--运算

++i(或i++)相当于i=i+1，而--i(i--)相当于i=i-1;

使用自增和自减运算符注意以下几点：

（1）只能用于变量，不能用于常量或表达式，如5++或(a+b)++都不合法

（2） ++i和i++的区别：作为表达式参加运算时， ++i先令i加1然后参加运算，而i++则先让i参加运算，然后再令i加1。

例如： int i=5,j; j=++i;

//i的值先增加1，然后再参与赋值运算，结果： i=6,j=6

再例如： int i=5,j;j=i++;

//i的值先参与赋值运算，然后i的值增加1；结果： i=6,j=5

2.4.3 类型转换

操作数的类型转换方式有两种：

- 2.4.3.1 隐式类型转换
  - 是指由C编译器按照某种预订的规则进行自动转换。
- 2.4.3.2 显示类型转换
  - 是指由编程员在程序中用类型转换运算符明确地指出转换。

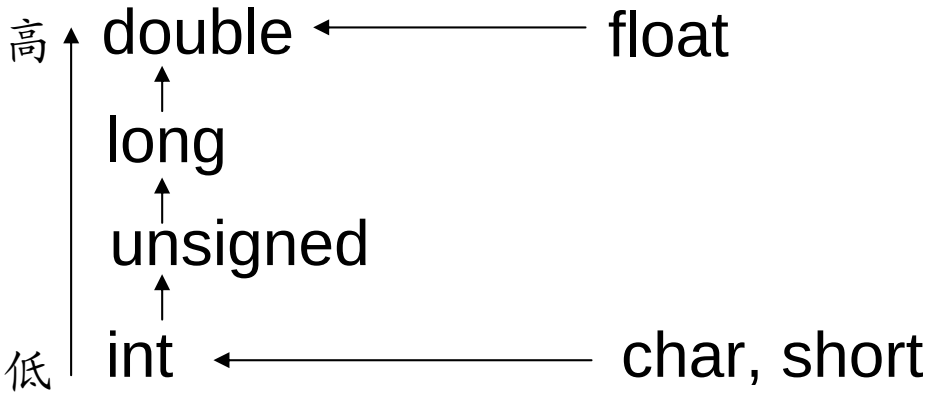
2.4.3.1 隐式类型转换

- 一个双目运算符两边的操作数的类型必须一致，才能进行运算操作，但C语言允许在一个表达式中的双目运算符两边出现不同数据类型的操作数。
- 运算时，C编译器会先对其中的一些操作数按约定的规则自动进行类型转换，使得双目运算符两边的操作数的类型一致，然后进行运算

隐式类型转换的规则：

（1）无条件的隐式类型转换：所有的char型和short型数据在运算前都必须转换为int型，所有的float型数据都必须转换成double型。即使双目运算符的两个操作数都是相同类型，也要进行类型转换。

（2）统一类型的隐式类型转换：如果双目运算符两边的操作类型不一致，则需要将其中类型较低的转换为较高的类型，然后基于同一类型进行运算。



2.4.3.1 隐式类型转换

- 例：f为float型变量，则算术表达式'A'+'B'-f求值时，不同数据类型的转换及运算顺序为：
- （1）进行'A'+'B'的运算：先将'A'和'B'都转换成int型（无条件转换），得到55和66，然后对这两个int型数据进行加法运算，结果为int型数据131。
- （2）进行131-f的运算：先将int型的131和float型的f都转换成double型，然后相减（其中，int到double型的转换是统一类型转换，float到double型的转换是无条件转换），得到的结果是double型。

2.4.3.2 显示类型转换

- 编程员在程序中借助强制类型转换运算符，人为地强制进行类型转换。
- 格式：(类型名)表达式
  - (double)t 将变量t的值强制转换成double型
  - (int)(x+y) 将表达式x+y的值强制转换成int型
  - (int)x%i 只对x的值强制转换为int型，再被i除取余

注意：

（1）从表示范围大的类型强制转换到表示范围小的类型，有时会丢失精度。例如float a=2.5,b=2，则表达式(int)a*b的值为4，而(int)(a*b)的值为5。

（2）无论是隐式类型转换，还是显示类型转换，都不会改变被转换的变量的值，转换得到的结果将存储在临时的存储单元中。

2.4.4 赋值运算符和赋值表达式

- 1、简单赋值运算符

  例如：a=4，把常量4赋值给变量a。

- 2、复合赋值运算符

  x%=10 //等价于x=x%10

  x*=x-y //等价于x=x*(x-y)

  x/=x+y //等价于x=x/(x+y)

  x+=y //等价于x=x+y

- 3、赋值表达式（难以理解，尽量避免使用）
  - a=b=5和a=(b=5)等价，都先求b=5的值，然后再赋值给a，a也是5。
  - a+=a-=a*a，假设a的初值是12，按右结合性，该表达式求解过程为：先计算a-=a*a的值，相当于a=a-a*a，值为-132。所以变量a的值为-132。接着再进行a+=-132的运算，相当于a=a+(-132)，因而最后变量a的值为-264。

2.4.4 赋值运算符和赋值表达式

- 4、赋值时的隐式类型转换

  如果赋值运算符两边数据类型不一致，则系统先将右边表达式值的类型自动转换成左边变量的类型，然后再进行赋值。

转换规则

不管赋值运算符右边的操作数是什么类型（较高或较低），一律自动转换为赋值运算符左边的变量的类型。

注意：

在不同类型数据之间赋值时，会出现意想不到的结果。

例如将一个浮点型数据5.99999赋值给一个整型变量i时，在转换过程中会舍掉小数部分，仅取整数部分。

2.4.5 逗号运算符和逗号表达式

- 1、逗号运算符
- 在C语言中，逗号“,”既是一个分隔符，又是一个运算符，即逗号运算符。它是所有C运算符中优先级别最低的，其结合性是从左往右。
- 2、逗号表达式
  - 语法格式：表达式1,表达式2,表达式3,…,表达式n
  - 求解过程：从左往右计算各个表达式的值，并且规定把最后一个表达式n的值作为整个逗号表达式的值。
  - 例如：

    逗号表达式“16+5,6+8”的值为14

    逗号表达式a=2*3,a*4的值为24

附件：课程教师和助教（2011-2012第2学期）

主讲教师：林子雨

单位：厦门大学信息科学与技术学院计算机科学系
办公地点：福建省厦门市思明区厦门大学海韵园
E-mail:ziyulin@xmu.edu.cn
个人主页：http://www.cs.xmu.edu.cn/linziyu

助教：林尚青

单位：厦门大学计算机科学系2010级硕士研究生
E-mail:lsq1015@qq.com
手机：15959206201

助教：赖明星

单位：厦门大学计算机科学系2011级硕士研究生
E-mail:joy_lmx@163.com
手机：18050056577

附件：课程FTP （2011-2012第2学期）

- FTP地址：ftp://218.193.53.74
- 用户名： stu_linziyu
- 密码：123456
- 目录：“下载教学内容”→”C语言”

附件：课程教材 （2011-2012第2学期）

- 《C语言程序设计（第2版）》
- 清华大学出版社，黄保和，江弋 编著
- 版次：2011年10月第2版
- ISBN:978-7-302-26972-4
- 定价：35元

Department of Computer Science, Xiamen University, February 19, 2012