



Xidian University

C语言程序设计

Lec 2 数据对象与计算





主要内容

- ✿ 引言
- ✿ 2.1 基本字符、标识符、关键字
- ✿ 2.2 数据、类型、变量
- ✿ 2.3 运算符、表达式与计算
- ✿ 小结





一个例子

- 编写程序求两直角边长度分别为 3.5米和 4.72 米的直角三角形的面积。

- 已知(输入)

- $a=3.5$

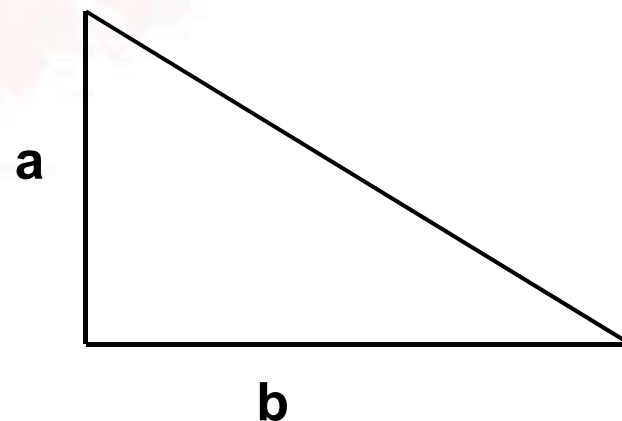
- $b=4.72$

- 计算

- $s=0.5 \times b \times a$

- 结果(输出)

- 输出s的值





例子转换为程序

❁ 需要解决的问题

- ❁ a, b, s在程序中用什么名字表示（标识符问题）？
- ❁ a, b, s在程序中用什么数据类型？
- ❁ a, b的值如何给定？（输入和赋值问题）
- ❁ 计算s的数学表达式如何转换成C语言表达式（运算符、优先级...）？
- ❁ 如何输出s的值？（输出问题）



示例程序

```
#include <stdio.h>
```

```
int main(){
```

```
    double a=3.5;
```

//一条直角边长

```
    double b=4.72;
```

//另一条直角边长

```
    double s=0;
```

//面积

```
    s=0.5*b*a;
```

//计算面积

```
    printf( "s=%lf\n", s);
```

//输出结果

```
    return 0;
```

```
}
```



2.1 基本字符、标识符、关键字

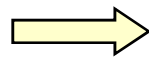




基本字符

- ❁ 数字字符: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- ❁ 大小写拉丁字母: a~z, A~Z;
- ❁ 其他一些可打印（可以显示）的字符，包括:
~!%&*()_ - + = { } [] : ; " ' < > , . ? / | \
- ❁ 特殊字符，如空格符、换行符、制表符等

中文字符能
否使用?



在注释中可以使用
在双引号内可以使用



基本字符

✿ 键盘上的特殊字符





名字（标识符）的构成

❖ 程序中的名字（表示一个程序对象）称为标识符

- ❖ 字母（包括下划线"_"）和数字字符的一个连续序列
- ❖ 不能有空白字符：Err or
- ❖ 第一个字符必须是字母或下划线"_"（注意下划线"_"与横线"-"的区别）
- ❖ 大小写敏感：NAME和name是不同的标识符



(demo_object.c)

标识符示例

合法标识符

abcd Beijing C_Programming _f2048
sin a3b06 xt386ex A_great_machine
Small_talk_80 FORTRAN_90

非法标识符

3set a[\$\$\$\$ sin(2 ::ab4==



关键字

- ❁ C 语言合法标识符中的一个特殊的小集合
- ❁ 作为**关键字的标识符**在程序里具有语言预先定义好的**特殊意义**，不能用于其他目的
- ❁ 不能作为普通的名字使用





C语言的32个关键字

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	union
unsigned	void	volatile	while



(demo_object.c)

标识符命名原则

- ❖ 尽量使用有意义的单词: name, age, book
- ❖ 第一个字符尽量不使用下划线:
_name, my_age, second_Book;
- ❖ 标识符由多个单词构成时可以每个单词首字母大写: numOfStudent, firstName, SecondBook;
- ❖ 标识符不宜太短, 也不宜太长, 可以适当使用简写
- ❖ 保持一致性



2.2 数据、类型、变量





数据与类型

- ❁ C语言的基本数据类型包括
 - ❁ 字符类型
 - ❁ 整数类型
 - ❁ 实数类型
- ❁ 程序中的每一个数据都**必定**属于某一种类型（基本类型或自定义类型）
- ❁ 每一种数据类型都有**表示方式**（**编码**）和**表示范围**
- ❁ 数据类型确定了数据对象支持的**运算**



字节

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

变量 a ←

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

 32767 ←

变量 b ←

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

 3 ←





(1) 整数类型和整数的表示

✿ 三种整数类型

- ✿ 长整数 `long int` 可以简写为 `long`
- ✿ 普通整数 `int`
- ✿ 短整数 `short int` 可以简写为 `short`

✿ 三种整数类型表示范围

- ✿ `long` \geq `int` \geq `short`, 具体范围取决于不同的实现

	<code>long</code>	<code>int</code>	<code>short</code>
字节数	4	4	2
表示范围	$-2^{31} \sim 2^{31}-1$	$-2^{31} \sim 2^{31}-1$	$-2^{15} \sim 2^{15}-1$



整数的书写形式 (`demo_int.c`)

❖ 十进制形式

❖ 例如：123456, 123, 1000

❖ 八进制形式

❖ 以0开头

❖ 例如：0123456, 0123, 01000

❖ 十六进制形式

❖ 以0x开头

❖ 例如：0x123456, 0x123, 0x1000



(2) 实数类型和实数表示

三种实数类型

单精度浮点数 `float` 1.123456

双精度浮点数 `double` 1.123456789012345

长双精度浮点数 `long double` 1.123456789012345678

三种实数类型的表示范围(IEEE标准)

	<code>float</code>	<code>double</code>	<code>long double</code>
字节数	4	8	10
有效位数	7(绝对保证6位)	16(绝对保证15位)	19
表示范围	$-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$	$-1.7 \times 10^{-308} \sim 3.4 \times 10^{308}$	$-1.2 \times 10^{-4932} \sim 3.4 \times 10^{4932}$



实数的书写形式

❁ 小数形式

❁ 例如：12000000.0 0.00001

❁ 指数形式

❁ 例如：1.2E7 1E-5

❁ 注意：以上形式表示的实数均被当做double类型，如果希望是float类型需要在后面加上F或f

❁ 例如：12000000.0f , 0.00001F , 1.2E7F , 1E-5F



(3) 字符类型和字符的表示

- ✿ 字符数据主要用于输入输出和文字处理
- ✿ 字符类型char（一个字节的整数）
 - ▣ unsigned char: 0~255
 - ▣ char: -128~127
 - ▣ ASCII 字符集：所有大小写英文字母、数字、各种标点符号字符，还有一些控制字符，一共128 个



字符的表示

❖ 1. 用单引号括起的单个字符

❏ 例如: 'A', 'a', '1'

❖ 2. 以ASCII 码表示

❖ 3. 以反斜线开头的转义字符用来表示特殊字符（对未定义的转义字符直接输出反斜线后的字符）

转义字符	'\n'	'\''	'\\'	'\"'
实际输出的字符	换行	单引号'	反斜线	双引号"



ASCII 码

- ❁ 在计算机中每一个拉丁文字字母用一个0~127的数字表示，这个数字称为该字母的ASCII码
 - ❁ 第0~32号及第127号(共34个)是控制字符或通讯专用字符，如控制符：LF（换行）、CR（回车）、FF（换页）、DEL（删除）、BS（退格）、BEL（振铃）等；通讯专用字符：SOH（文头）、EOT（文尾）、ACK（确认）等；
 - ❁ 第33~126号(共94个)是字符，其中第48~57号为0~9十个阿拉伯数字；65~90号为26个大写英文字母，97~122号为26个小写英文字母，其余为一些标点符号、运算符号等。例如：65（对应字符'A'），97（对应字符'a'），49（对应字符'1'）

Ctrl	十进制	十六进制	字符	代码	十进制	十六进制	字符	十进制	十六进制	字符	十进制	十六进制	字符
^@	0	00		NUL	32	20	!	64	40	@	96	60	'
^A	1	01		SOH	33	21	!	65	41	A	97	61	a
^B	2	02		STX	34	22	"	66	42	B	98	62	b
^C	3	03		ETX	35	23	#	67	43	C	99	63	c
^D	4	04		EOT	36	24	\$	68	44	D	100	64	d
^E	5	05		ENQ	37	25	%	69	45	E	101	65	e
^F	6	06		ACK	38	26	&	70	46	F	102	66	f
^G	7	07		BEL	39	27	'	71	47	G	103	67	g
^H	8	08		BS	40	28	(72	48	H	104	68	h
^I	9	09		HT	41	29)	73	49	I	105	69	i
^J	10	0A		LF	42	2A	*	74	4A	J	106	6A	j
^K	11	0B		VT	43	2B	+	75	4B	K	107	6B	k
^L	12	0C		FF	44	2C	,	76	4C	L	108	6C	l
^M	13	0D		CR	45	2D	-	77	4D	M	109	6D	m
^N	14	0E		SO	46	2E	.	78	4E	N	110	6E	n
^O	15	0F		SI	47	2F	/	79	4F	O	111	6F	o
^P	16	10		DLE	48	30	0	80	50	P	112	70	p
^Q	17	11		DC1	49	31	1	81	51	Q	113	71	q
^R	18	12		DC2	50	32	2	82	52	R	114	72	r
^S	19	13		DC3	51	33	3	83	53	S	115	73	s
^T	20	14		DC4	52	34	4	84	54	T	116	74	t
^U	21	15		NAK	53	35	5	85	55	U	117	75	u
^V	22	16		SYN	54	36	6	86	56	V	118	76	v
^W	23	17		ETB	55	37	7	87	57	W	119	77	w
^X	24	18		CAN	56	38	8	88	58	X	120	78	x
^Y	25	19		EM	57	39	9	89	59	Y	121	79	y
^Z	26	1A		SUB	58	3A	:	90	5A	Z	122	7A	z
^[27	1B		ESC	59	3B	;	91	5B	[123	7B	{
^\	28	1C		FS	60	3C	<	92	5C	\	124	7C	
^]	29	1D		GS	61	3D	=	93	5D]	125	7D	}
^^	30	1E	▲	RS	62	3E	>	94	5E	^	126	7E	~
^-	31	1F	▼	US	63	3F	?	95	5F	-	127	7F	ÿ

控制符:

LF (换行)、
CR (回车)、
FF (换页)、
DEL (删除)、
BS (退格)、
BEL (振铃) 等

通讯专用字符:

SOH (文头)、
EOT (文尾)、
ACK (确认)

* ASCII 代码 127 拥有代码 DEL。在 MS-DOS 下, 此代码具有与 ASCII 8 (BS) 相同的效果。
DEL 代码可由 CTRL + BKSP 键生成。



字符的特殊性

- ✿ 字符数据与标识符不同, 例如: **x** 和 **'x'**

int x=1;

char a='x'; 等效于: **char a=120;**

- ✿ 数字字符和数不同, 例如: **1** 和 **'1'**

int x1=1;

char a='1'; 等效于: **char a=49;**

- ✿ 程序中表示一个字符需要用单引号, 但从键盘输入字符时不需要加单引号



字符串

- ❖ 字符串不是C语言的一种数据类型
- ❖ 用双引号括起的一系列字符：
 - ❖ "CHINA" "Beijing" "University" "Welcome\n"
- ❖ 其本质是字符数组
- ❖ 主要用于输入输出
 - ❖ `printf("Hello C Program!\n");`
- ❖ 字符串中间不能换行

"Hello C Program"



"Hello C
Program"





变量声明

- ❁ **变量**：用于存储程序的输入数据或计算结果的存储单元
 - ❁ 变量值在程序运行时可以改变
- ❁ **变量声明**：给存储单元定义一个名称及类型，便于程序中引用
 - ❁ 必须是合法标识符：**字母或 '_' 开头，字母数字组合**
 - ❁ 必须有确定数据类型：**int, float, double, char ...**
 - ❁ 可以在一条语句中定义多个同类型变量
 - ❁ 可以在变量声明时赋值



示例程序中的变量

```
#include <stdio.h>
```

```
int main(){
```

```
    double a=3.5, b=4.72; //表示两条直角边的变量
```

```
    double s=0;           //表示面积的变量
```

```
    s=0.5*b*a;            //计算面积
```

```
    printf( "s=%lf\n", s); //输出结果
```

```
    return 0;
```

```
}
```




变量基本操作

✿ 赋值操作——改变变量当前的值

- ✿ 方法：使用赋值表达式（由操作符“=”构成的表达式）

- ✿ $a=3.5$

- ✿ $b=4.72$

- ✿ $s=0.5*b*a$

✿ 取值操作——获得变量现在的值

- ✿ 方法：直接引用变量名称

- ✿ $s=0.5*b*a$

- ✿ `printf("s=%d\n", s);`





示例程序中的变量使用

```
#include <stdio.h>

int main(){
    double a=3.5, b=4.72; //给变量a, b赋值
    double s=0;           //表示面积的变量

    s=0.5*b*a; //取变量a, b的值计算面积, 赋值给变量s
    printf( "s=%lf\n", s); //取变量s的值用于输出

    return 0;
}
```



程序的核心是**存取数据**，要理解一个程序最重要的就是了解程序运行过程中数据的变化





2.3 运算符、表达式与计算





算术运算符

运算符	使用形式	意义	适用数据类型
+	一元或二元运算符	一元表示正号，二元表示加法： $+1$, $2+3$	所有基本数据类型
-	一元或二元运算符	一元表示负号，二元表示减法： -1 , $3-2$	所有基本数据类型
*	二元运算符	乘法运算： $3*5$, $a*b$	所有基本数据类型
/	二元运算符	除法运算： $6/3$	所有基本数据类型
%	二元运算符	模运算（求余数）： $8\%3$	整数类型



算术运算符示例 (demo_operator.c)

```
int main()
{
    double a=3.0,b=2.0;
    int c=4,d=3;

    printf("-a=%f\n", -a );
    printf("a+b=%f\n", a+b );
    printf("a-b=%f\n", a-b );
    printf("a*b=%f\n", a*b );
    printf("a/b=%f\n", a/b );
    printf("c/d=%d\n", c/d );
    printf("c%%d=%d\n", c%d );
}
```

输出结果:

```
-a=-3.000000
a+b=5.000000
a-b=1.000000
a*b=6.000000
a/b=1.500000
c/d=1
c%d=1
```



算术表达式

- 由计算对象（例如数值的文字量、变量、函数调用等）、算术运算符及圆括号构成，其基本形式与数学上的算术表达式类似

- $-(28 + 32) + (16 * 7 - 4)$

- $25 * (3 - 6) + 234$

- $a * b + c / 2$!! Error (C ÷ 2)

- $a * \sin(c * 3.1416 / 180)$



表达式求值

❁ 表达式的计算过程

- ❁ **优先级**—不同运算符哪个先算，哪个后算？
- ❁ **结合性**—相同优先级的运算符哪个先算，哪个后算？
- ❁ **求值顺序**—二元运算符的两个运算对象哪个先算，哪个后算？





表达式求值

❁ 优先级

- ❁ 一元运算符优先级最高
- ❁ 先乘除后加减
- ❁ 括号中的表达式先计算

运算符	一元+和-	* / %	二元+和-
优先级	高	中	低

$$5 / 4 + 4 * 6 / 2 = ? \quad \quad \quad 13$$

$$(((2 + 6) * 4) / (3 + 5)) = ? \quad \quad \quad 4$$

当不确定运算符优先级时，用括号来指定运算顺序是避免错误的最好方法



表达式求值

结合性

- 确定具有相同优先级的运算符相邻出现时表达式的计算顺序。
- C语言规定一元算术运算符自右向左结合；二元算术运算符自左向右结合，优先级相同时左边的运算符先计算。

$$4 * 7 / 2 \longrightarrow (4 * 7) / 2 \longrightarrow 14$$

$$-+-8 \longrightarrow (-(+(-8))) \longrightarrow 8$$



表达式求值

❁ 求值顺序

- ❁ 二元运算符的两个运算对象的计算顺序问题
- ❁ C语言没有明确规定，取决于编译器的实现

$$(5+8) * (6 + 4)=?$$

已知 $a=1, b=2$
 $(a+b)*(++a)=?$

(Demo_Priority.c)

程序里不应该写对求值顺序敏感的表达式



计算和类型

$5 / 4 + 4 * 6 / 2 = ?$	13	Error: 13.25
$1 / 3 * 3 = ?$	0	Error: 1
$1 * 3 / 3 = ?$	1	
$1/3.0 * 3 = ?$	1.0	
$(\text{int})(3.3 * 2.2) + 4 = ?$	11	

- ❖ 类型对计算的限制
- ❖ 混合类型计算和自动（隐式）类型转换
- ❖ 强制（显式）类型转换



计算和类型

✿ 类型对计算的限制

- 1. 两个相同类型数据使用二元运算符计算得到的结果类型相同

- $5 / 4 + 4 * 6 / 2 = 13$

- $1 / 3 * 3 = 0$

- $1.0 / 3.0 * 3.0 = 1.0$



计算和类型

❁ 类型对计算的限制

- ❁ 2. 计算结果超出类型标识范围导致“溢出”
(Demo_overflow.c)

```
#include <stdio.h>
int main(){
    short a=32767;
    a = a+1;
    printf("a=%d\n",a); // 输出: a=-32768
    return 0;
}
```





计算和类型

❁ 混合类型计算和自动（隐式）类型转换

❁ 两个不同类型数据使用二元运算符形成混合类型计算

- C语言里**没有混合类型**的算术运算
- C语言需要将混合类型计算中的某个运算对象转换为另一个运算对象的类型再计算： **$3*2.0$**
- 转换原则是**将表示范围小的类型转换为表示范围大的类型以避免丢失精度**



计算和类型

表示范围	小						大
类型	short	int	long	float	double	long double	

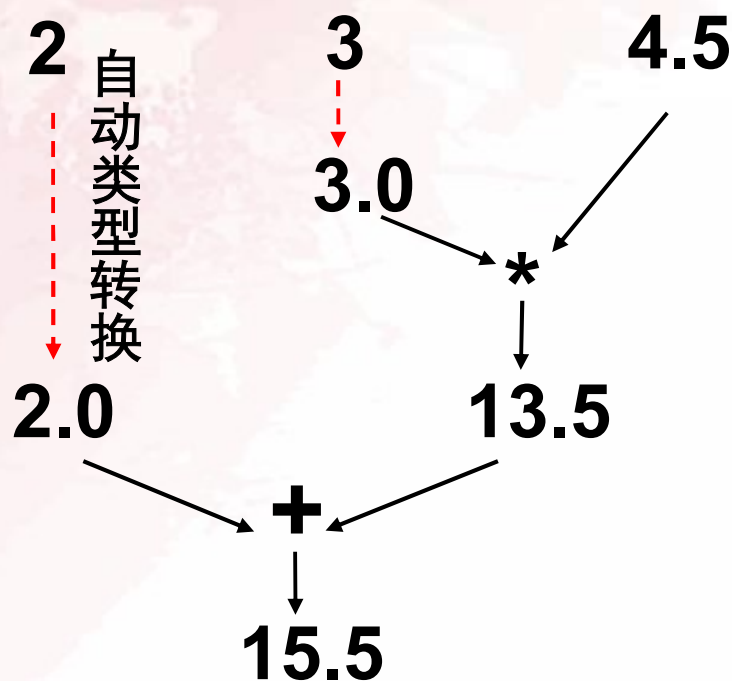
浮点数类型转换为整数时直接去掉小数部分



计算和类型

混合类型计算和自动（隐式）类型转换

■ $2 + 3 * 4.5 = 15.5$





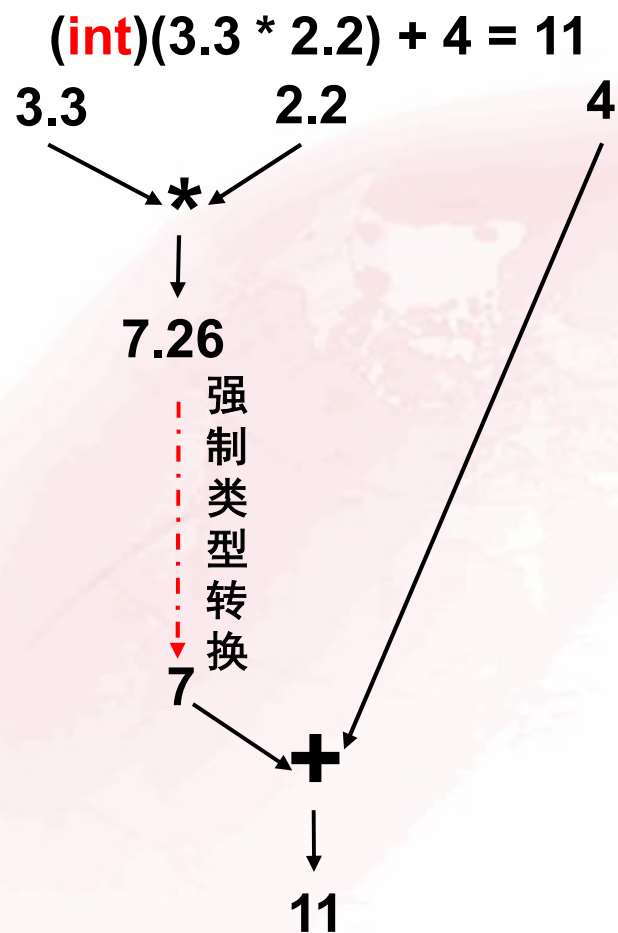
计算和类型

❁ 强制（显式）类型转换

- ❁ 为了计算需要: `int a=3; float b=a; float b=(float) a;`
- ❁ 转换方式: (类型名) 表达式
- ❁ 转换可能丢失精度
- ❁ 任何类型均可相互转换, 但转换之后不一定有意义
- ❁ 强制类型转换是一元运算符, 其优先级和其他一元运算符相同 (高于二元运算符)
 - `(int) 3.3 * 2.2 + 4 = 10.6`
- ❁ 类型转换不改变原来的值
 - `float a=3.3;`
 - `int b=(int) a; //b=3, 但a的值不会改变`



计算和类型





输出计算结果

❁ 格式化输出函数printf

❁ C语言标准库函数: `stdio.h`

❁ 函数声明: `printf(char *format, ...);`

❁ 调用形式: `printf(格式字符串, 其他参数);`

控制输出格式

- 输出类型
- 输出精度
- 对齐方式

准备输出的值

- 0个或多个
- 所有基本类型变量或常量
- 表达式或其他函数调用结果



输出计算结果

(demo_printf-f-lf.c)

❁ printf函数的格式字符串

- ❁ 双引号括起来的字符串: printf(“a=%d”, 10);
- ❁ 用%引导格式字符
 - %d - 以十进制输出带符号整数 (decimal)
 - %o - 以八进制输出无符号整数 (octal)
 - %x - 以十六进制输出无符号整数 (hexadecimal)
 - %f - 以小数形式输出float/double型浮点数, 默认小数点后输出6位有效数字 (float/double)
 - %lf - 以小数形式输出long double型浮点数 (long double)
 - %e - 以科学计数法输出float/double型浮点数, 默认小数点后输出6位有效数字 (float/double)
 - %c - 输出一个字符 (character)
 - %s - 输出一个字符串 (string)
- ❁ 格式字符串中的其他部分原样输出 (转义字符除外)



输出计算结果

❁ 示例 (demo_printf1.c)
(demo_printf-f-lf.c)

输出结果:

a=256, b=-180

a=400, b=377777777514

a=100, b=ffffff4c

c=1234.567000, d=2356.435791

c=1.234567e+003, d=2.356436e+003

ch1=A, ch2=a

ch1=65, ch2=97

I love c programming !

```
#include <stdio.h>
```

```
int main(){
```

```
int a=256,b=-180;
```

```
double c=1234.567;
```

```
float d=2356.4359f;
```

```
char *s="c programming";
```

```
char ch1='A',ch2='a';
```

```
printf( "a=%d, b=%d\n", a, b);
```

```
printf( "a=%o, b=%o\n", a, b);
```

```
printf( "a=%x, b=%x\n", a, b);
```

```
printf( "c=%f, d=%f\n", c, d);
```

```
printf( "c=%e, d=%e\n", c, d);
```

```
printf( "ch1=%c, ch2=%c\n", ch1,ch2);
```

```
printf( "ch1=%d, ch2=%d\n", ch1, ch2);
```

```
printf( "I love %s !\n", s);
```

```
return 0;
```

```
}
```




输出计算结果

✿ 使用printf需要注意的几个问题

- ✦ 格式字符串中没有%引导的格式控制字符时，不需要其他参数，直接输出格式字符串：`printf("hello");`
- ✦ 格式字符串中有%引导的格式控制字符时，其他参数的数量以及类型必须和格式控制字符一致（类型不一致时**不会**进行类型转换，而会将实际传入的值当作需要的类型来理解，因此可能出现非预期结果。
- ✦ 关于精度等问题可以参考printf函数手册



输出计算结果

❖ 示例（demo_printf2.c）

```
#include <stdio.h>
```

```
int main(){
```

```
    int a=256;
```

```
    float d=2356.4359f;
```

```
    printf( "a=%f, d=%d \n", a, d); //???? 不一致，出错！
```

```
    printf( "a=%f,d=%d\n", (float)a, (int)d); //强制转换可以
```

```
    return 0;
```

```
}
```

类型不一致时**不会进行类型转换**，而会将实际传入的值当作需要的类型来理解，因此**可能出现非预期结果!!!!**



如何得到用户输入

❁ 求三角形面积的程序存在的问题

❁ a和b都是固定值，如果修改a或b的值程序需要重新编译

```
#include <stdio.h>
int main(){
    double a=3.5;           //一条直角边长
    double b=4.72;          //另一条直角边长
    double s=0;              //面积
    s=0.5*b*a;               //计算面积
    printf( "s=%lf\n", s);   //输出结果

    return 0;
}
```

能否让用户输入a和b的值再计算面积？



从键盘输入数据

❖ 格式化输入函数scanf

❖ C语言标准库函数

❖ 函数声明: `scanf (char *format, ...);`

❖ 调用形式: `scanf (格式字符串, 其他参数);`

控制输入数据格式
(类型, 精度等)
的字符串, 以%引导

接受输入值的**变量地址**
(在普通变量前加
取地址运算符**&**可得到
变量地址)



scanf函数的格式字符串

格式控制字符	接受输入值的变量类型	要求的实际输入
%d	int	十进制数
%ld	long	十进制数
%f	float	十进制数，可以有小数点或指数
%lf	double	十进制数，可以有小数点或指数
%Lf	long double	十进制数，可以有小数点或指数
%c	char	一个字符



scanf示例1 (demo_scanf1.c)

```
int x;  
float y;  
double z;  
char c;  
scanf("%d %f %lf %c", &x, &y, &z, &c);  
printf("x=%d, y=%f, z=%f, c=%c", x, y, z, c);
```

用户从键盘输入以下内容 (□表示空格) :
234□2252. 18□220. 4□A

输出结果:
x=234, y=2252.179932, z=220.400000, c=A

❁ 注意事项

- ❁ 接受输入值的参数是**变量地址**
- ❁ **必须要按下回车后才结束输入**
- ❁ 输入多个数据时，格式字符之间一般用**空格**分隔或者**不**分隔，但输入数据要用空格分隔
- ❁ 一旦在格式字符串中用了**别的字符**分隔，输入时也要用相同字符分隔



scanf 示例2

```
int x;  
float y;  
double z;  
char c;  
scanf("%d ,%f, %lf, %c", &x ,&y ,&z, &c);  
printf("x=%d, y=%f, z=%f,c=%c", x, y, z,c);
```

用户从键盘输入以下内容：
234, 2252.18, 220.4, A

输出结果：
x=234, y=2252.179932, z=220.400000,c=A

(demo_scanf2.c)

对比：
(demo_printf-f-lf.c)



数学函数及其使用



- ❖ 另一种求任意三角形面积的方法是已知两边和夹角求三角形的面积。

- ❖ 已知(输入)

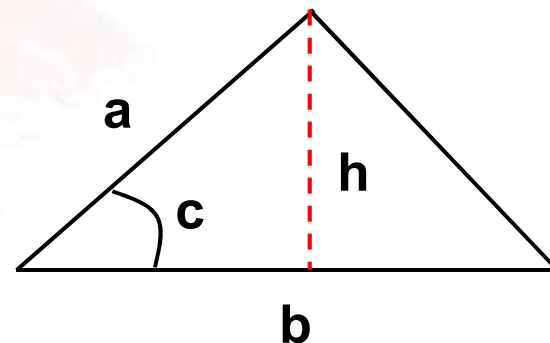
- $a=3.5$
- $b=4.72$
- $c=37$

- ❖ 计算

- $h=a*\sin(\theta)$
- $s=0.5*b*h=0.5 * b * a * \sin(c)$

- ❖ 结果(输出)

- 输出s的值



sin值怎么求?



数学函数及其使用

- ❖ C语言的**扩展函数库**中有一类函数可用于求解数学函数：**math.h**（**对应的位置**）
- ❖ 使用函数需要的准备工作
 - ❑ 知道函数声明所在的头文件
 - ❑ 知道函数名字
 - ❑ 知道**函数使用形式（入口参数）**
 - ❑ 知道函数产生的**结果（返回值）类型**，该结果可用于给变量赋值或作为表达式的一部分

} 函数声明



数学函数及其使用

✿ 数学函数使用示例

```
#include <math.h> // 数学函数头文件，包含了sin的声明
#include <stdio.h> // I/O函数头文件，包含了printf的声明

int main(){
    ...
    h=a*sin(c*3.1416/180); //使用sin函数计算角度c的正弦值
    ...
    return 0;
}
```



数学函数及其使用

❖ `double sin(double rad)` ;

- ❑ `sin`函数用来计算一个弧度的正弦值
- ❑ `sin`函数要求传入参数是一个表示弧度的`double`类型值
- ❑ `sin`函数返回`double`类型的正弦值

❖ 其他常用数学函数

- ❑ `double sqrt(double x)`, 求 x 的平方根
- ❑ `double pow(double x, double y)`, 求 x^y
- ❑ `double fabs(double e)`, 求 e 的绝对值
- ❑ `double log(double)`, 求以 e 为底的对数
- ❑ `double log10(double)`, 求以10为底的对数



(demo_math.h)

数学函数及其使用

- 使用函数时注意参数类型匹配，即函数声明中要求的参数类型和实际传入的数据类型应该一致

```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main () {
```

```
    printf("%f\n",
```

```
        sin(1) + sin(1/2) +
```

```
        sin(1/3) + sin (1/4) +
```

```
        sin(1/5) + sin(1/6) +
```

```
        sin(1/7) + sin(1/8) +
```

```
        sin(1/9) + sin(1/10) );
```

```
    return 0;
```

```
}
```

错误



$$\sum_{n=1}^{10} \sin(1/n) = ?$$

正确



```
#include <stdio.h>
```

```
#include <math.h>
```

```
int main () {
```

```
    printf("%f\n",
```

```
        sin(1.0) + sin(1.0/2) +
```

```
        sin(1.0/3) + sin (1.0/4) +
```

```
        sin(1.0/5) + sin(1.0/6) +
```

```
        sin(1.0/7) + sin(1.0/8) +
```

```
        sin(1.0/9) + sin(1.0/10) );
```

```
    return 0;
```

```
}
```



小结

✿ 需要掌握的内容

- ✦ C语言字符集
- ✦ 标识符
- ✦ 基本数据类型
- ✦ 运算符、表达式
- ✦ 变量定义及使用
- ✦ 格式化输入函数scanf，格式化输出函数printf
- ✦ 数学函数使用



小结

常用程序模式

```
#include <stdio.h>
int main() {
    //输入数据
    scanf("...",...);
    ...
    //计算
    ... // 计算表达式写在这里

    //输出结果
    printf("... ..", ...);
    ...
    return 0;
}
```