第2章 数据类型、运算符与表达式

本章介绍基本数据类型的定义与使用，输出/输入函数的使用，并能编制顺序结构的简单程序；介绍算术运算符和算术表达式、赋值运算符和赋值表达式以及数据类型转换规则，从而进一步提高顺序结构程序的设计能力。

主要内容

1. 标识符、变量与常量
2. 基本数据类型
3. 输出、输入函数
4. 算术运算
5. 赋值运算
6. 数据类型转换

变量使用举例

例：从键盘输入任意两个整数，求它们的和与积。

```
#include "stdio.h"
 main()
{   int a,b,sum，product;
  printf("Please input two integers:\n");
  scanf("%d,%d",&a,&b);/*格式输入函数*/
  sum=a+b;
  product=a*b;
  printf("sum =%d\n",sum);
  printf("product =%d\n",product);
}
```

定义整型变量

给变量赋值

计算

输出

标识符

- 有许多需要由程序员命名的对象(变量名、符号常量名、函数名、数组名、文件名等)，这些对象统称为标识符。
- C语言的标识符规定：
  - 只能由英文字母、下划线、数字组成，且只能用字母和下划线开头。
  - 变量名区分大小写，a和A， p和P是不同的变量。通常变量都用小写，以增加程序的可读性。
  - 长度建议不要超过8个。
  - 取名字时，最好能见名思义，如sum、name、age等分别用来存储总和、名字、年龄的数值。

标识符

关键字：是指在C语言中已预先定义具有特定含义的标识符，通常也称为保留字，编程者不得再重新命名另作他用。不能使用系统保留字作为标识符，C语言有32个关键字，如：int，short，break等。

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			

下列哪些是合法的标识符？

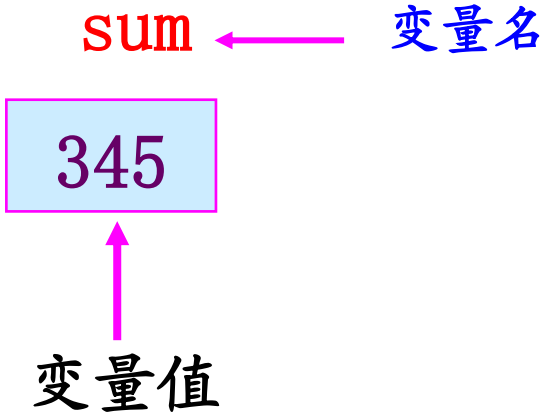
a　　str2　　add100　　student　　area
class_5　　a_1

3th　　#xyz　　“m+n”　　person name　　int

常量与变量

数据有两种形式：

- 常量：具有固定值，且值不能改变，如345，3.5，‘a’，“ok”



- 变量：其值可以改变，是一个内存空间。每一个变量都有一个名字

- 变量必须先定义后使用

变量的命名规则：以字母或下划线开头，由数字、字母、下划线构成。最多32个字符，多余不识别，一般小写。

变量声明与赋值

变量要先声明（定义），后使用，定义的任务包括：变量的类型（int ,float),变量的名字以及初值。如:

```
int a,sum=0;
```

对变量的基本操作是赋值，通过赋值运算符（=）可以改变变量的值，如：

```
int x;
x=3;/*  3存入x变量，x变量的值为3  */
x=5; /*  5存入x变量，x变量的值为5 */
x=x+1;
x=x*x; /*  该语句执行后，x的值为多少？*/
```

变量声明与赋值

变量必须先定义后使用。C语言允许在定义变量时对变量进行初始化，即对变量赋初值。

例如:

```
int a=3;
float f=3.56, x=1.0;
char c= 'a' ;
int b,c,d=5,a;
int x=y=0;
b=c=8;
```

不能重复定义

不能连写

变量的动态特性

有程序段:

……

```
int a,b;
a=8; b=9;     /*a的值为8，b的值为9*/
a=b+1;        /*a的值变为10*/
a=b;          /*a的值变为9*/
```

……

- 存储器的存取特点是：取之不尽，一存就变。
- 即变量中的值可以反复读取，其值不会改变。而把一个新值赋给变量时，变量中原来的值就被新值所替代。

常量的使用

其值不能改变的量，称为常量，用标识符代表的常量称为符号常量。

例1_6 已知圆的半径，求圆的面积（符号常量）

```
#include "stdio.h"
#define PI 3.14/*PI为符号常量*/
main()
{
 float area1,area2;
 area1=2*2*PI;
 area2=5*5*PI;
 printf("area1=%f,area2=%f\n",area1,area2);
}
```

程序的执行结果：

area1=12.560000,area2=78.500000

常量与变量

```
#define PRICE  30
main()
{ int num,total;
  float v ,r,h;
  num=10;
  total=num*PRICE;
  printf("total=%d",tatal);
  r=2.5;
  h=3.2;
 v=3.14159*r*r*h;
  printf("v=%f",v);
}
```

符号常量

变量先定义后使用

名字常用大写

常量

变量

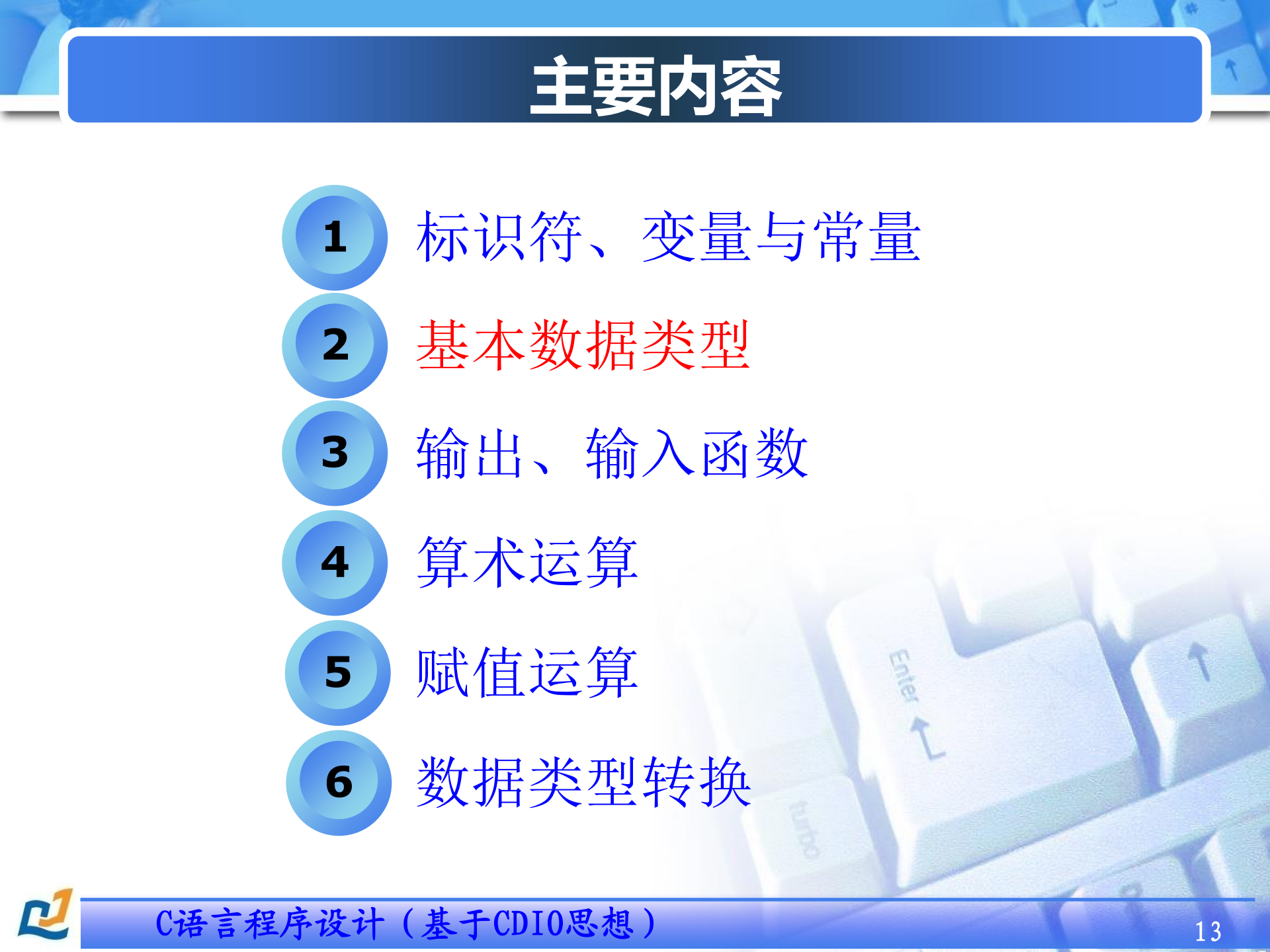
变量名：
以字母或下划线开头，由数字、字母、下划线构成。最多31个字符，多余不识别，一般小写。

常量：
程序执行中其值不会改变的量

变量：
程序执行中其值可改变的量

主要内容

1. 标识符、变量与常量
2. 基本数据类型
3. 输出、输入函数
4. 算术运算
5. 赋值运算
6. 数据类型转换



数据类型

数据是程序操作的对象。C语言根据数据不同性质和用途将其分为不同的数据类型。各种数据类型具有不同的存储长度、取值范围及允许的操作。

本讲只介绍基本类型

整型数据integer

1．整型常量：有三种形式的整型常量：

- 与习惯相同的十进制整数。如：46, -23, 0
- 以数字0开头的八进制整数。如：0456表示八进制数456, 即$(456)_8=4*8^2+5*8^1+6*8^0$, 等于十进制的302。
- 以数字0和x开头的十六进制整数。如：0x456表示十六进制整数456，即$(456)_{16}=4*16^2+5*16^1+6*16^0$，等于十进制的1110

2．整型变量

不同类型变量的字长（在内存中所占用的空间大小）取决于C编译器。如int在TC中就是16位（2字节）；在VC++中是32位（4字节）。

整型数据家族包括short int、int和long int，并且都分为signed和unsigned型。

整型数据integer

类　　型	字 节 数		取　值　范　围	
	TC	VC++	DOS16(TC)	Win32(VC++)
int	2	4	-2^{15} ~ $(2^{15}-1)$	-2^{31} ~ $(2^{31}-1)$
unsigned int	2	4	0 ~ $(2^{16}-1)$	0 ~ $(2^{32}-1)$
short [int]	2	2	-2^{15} ~ $(2^{15}-1)$	-2^{15} ~ $(2^{15}-1)$
unsigned short [int]	2	2	0 ~ $(2^{16}-1)$	0 ~ $(2^{16}-1)$
long [int]	4	4	-2^{13} ~ $(2^{31}-1)$	-2^{31} ~ $(2^{31}-1)$
unsigned long [int]	4	4	0 ~ $(2^{32}-1)$	0 ~ $(2^{32}-1)$

int型数的表示范围：二进制　16bit（2Byte）

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

int型整数的最大值

$=2^{15}-1=32767$

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

int型整数的最小值

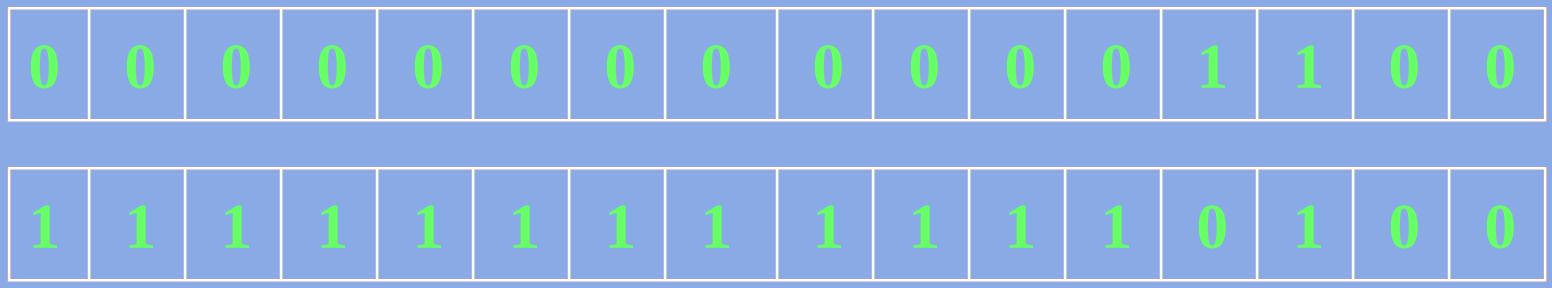
$=-2^{15}=-32768$

int型表示数的范围：－32768～32767

注意：使用中要防止数据溢出

数据的存储形式

有符号整数12与-12和存储形式(补码形式）：




```
main()
{
        int x=-1;
        printf("x=%d",x);
        printf("x=%u",x);
}
```

运行结果：

x=-1

x=65535

```
main( )
{ short int i;
  i= 1;
  i=i*2;  printf("\n i=%d",i);
  i=i*3;  printf("\n i=%d",i);
  i=i*4;  printf("\n i=%d",i);
  i=i*5;  printf("\n i=%d",i);
  i=i*6;  printf("\n i=%d",i);
  i=i*7;  printf("\n i=%d",i);
  i=i*8;  printf("\n i=%d",i);
  i=i*9;  printf("\n i=%d",i);
  i=i*10; printf("\n i=%d",i);
}
```

实际运行结果

```
i=2
i=6
i=24
i=120
i=720
i=5040
i=-25216
i=-30336
i=24320
```

产生数据溢出

上溢

下溢

整型数据integer

例: 以下程序有什么错误？请改正。

```
#include "stdio.h"
main()
{   short x,y,product;
    x=200;
    y=300;
    product=x*y;
    printf("%d\n",product);
}
```

实型（浮点型）数据

1. 实型常量

①小数形式。它由数字和小数点组成（必须有小数点）。如：12.3 、0.12 、456. 、124.0、 0.0等是合法的小数形式，而1/2是表达式，不是实数。

②指数形式（科学计数法）。用指数形式输出一个实数时，按规范化的形式输出，即小数点左边有且只有一位非零的数字。如：2.478e2、3.099e5。

注意e(E)前必须有数，e(E)后的指数必须为整数，如：e2、 2.1e3.5 、.e3 、e等都是非法的实数 。

指数： 1e3　　1.8e-3　　-123e-6　　-.1e-3　　.1E10

等价于:1000　　$1.8*10^{-3}$　　$-123*10^{-6}$　　$-0.1*10^{-3}$　　$0.1*10^{-10}$

实型（浮点型）数据

2. 实型变量

单精度型（float）、双精度型（double）、长双精度型（long double）。

特别要注意的是，long double的长度，不同的C编译环境，对它分配的长度有8、10、12不等

类　型	字　节　数		取　值　范　围	
	TC	VC++	DOS16（TC）	Win32（VC++）
float	4	4	10^{-37} ~ 10^{38}	10^{-37} ~ 10^{38}
double	8	8	10^{-307} ~ 10^{308}	10^{-307} ~ 10^{308}
long double	10	8	10^{-4931} ~ 10^{4932}	10^{-307} ~ 10^{308}

实型（浮点型）数据

很小的浮点数参与大数的加减等运算，往往会被忽略不计。

例：很大的浮点数与较小数参与的加减运算

```
#include "stdio.h"
main()
{ float a;
  a=123456.789e5;
printf("%f\n",a);
  a=a+20;
printf("%f\n",a);
}
```

程序运行结果如下：

```
12345678848.000000
12345678848.000000

--------------------------------
Process exited after 0.027
请按任意键继续. . .
```

字符型数据

1. 字符型常量

- 用单引号括起来的一个字符，如：如'a'、'+'、'2'、' '。
- 以‘\’开头的特殊字符(转义字符)，‘\n’、‘\t’，有的转为字符自身(‘\\’)，用转义字符可输出任何用ASCII码表示的字符,如‘\141’代表 ‘a’

2、字符型变量

用char定义的变量就是字符型变量,这种变量只有一个字节存储空间,存储字符型常量的ASCII码值,因此，整数在7位二进制表示的范围内（0-127），与字符型数据通用。

字符型数据举例

```
main()
{char c1,c2;
 int x;
 c1=97;
 c2= 'b' ;
 x=c1+2
 printf(" %c %d\n" ,c1,c2)
 printf ( " %c %d" ,x,x ) ;
}
```

- 字符数据和整型数据之间可以通用，可以按字符形式输出，也可以按整型输出。
- 字符数据与整型数据可以互相赋值。

运行结果为：

a 98
c 99

限制：整数在0----127之间才可以通用

转义字符

字符形式	功能	使用举例
\a	响铃功能	
\0	字符串结束标志	
\n	换行(ASCII码为10)	printf("\n");
\t	横向跳格	printf("\t");
\b	退格	
\r	回车(ASCII码为13)	
\f	走纸换页	
\\	字符\(ASCII码为92)	’\\’ 表示字符 \
\’	单引号	’\’’ 表示字符 ’
\”	双引号	’\”’ 表示字符 ”
\ddd	用8进制表示字符	’a’=’\141’ ’A’=’\101’ 换行符 ’\12’
\xhh	用16进制表示字符	’a’=’\x61’ ’A’=’\x41’ 换行符 ’\xa’

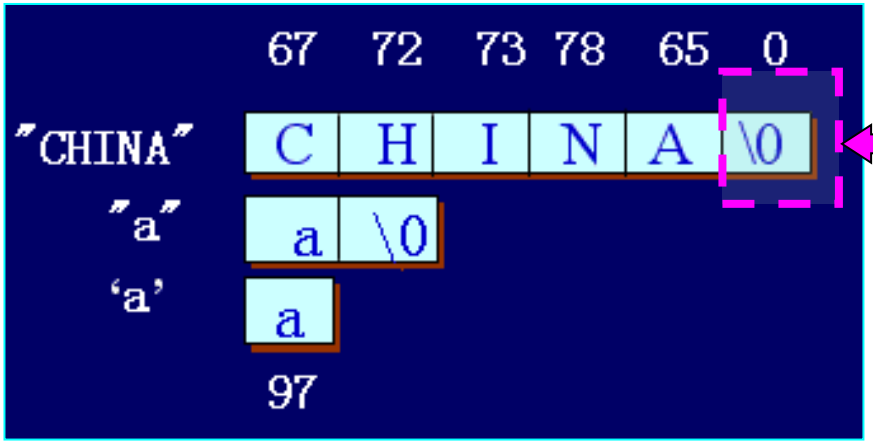
字符串常量

用双引号括起来的字符序列是字符串常量，如：“CHINA”、“a”、“2”等，字符串常量内部存储时以‘\0’结尾；

C语言中没有字符串变量，字符串常量要存储到字符数组中。

‘a’ | 97 |

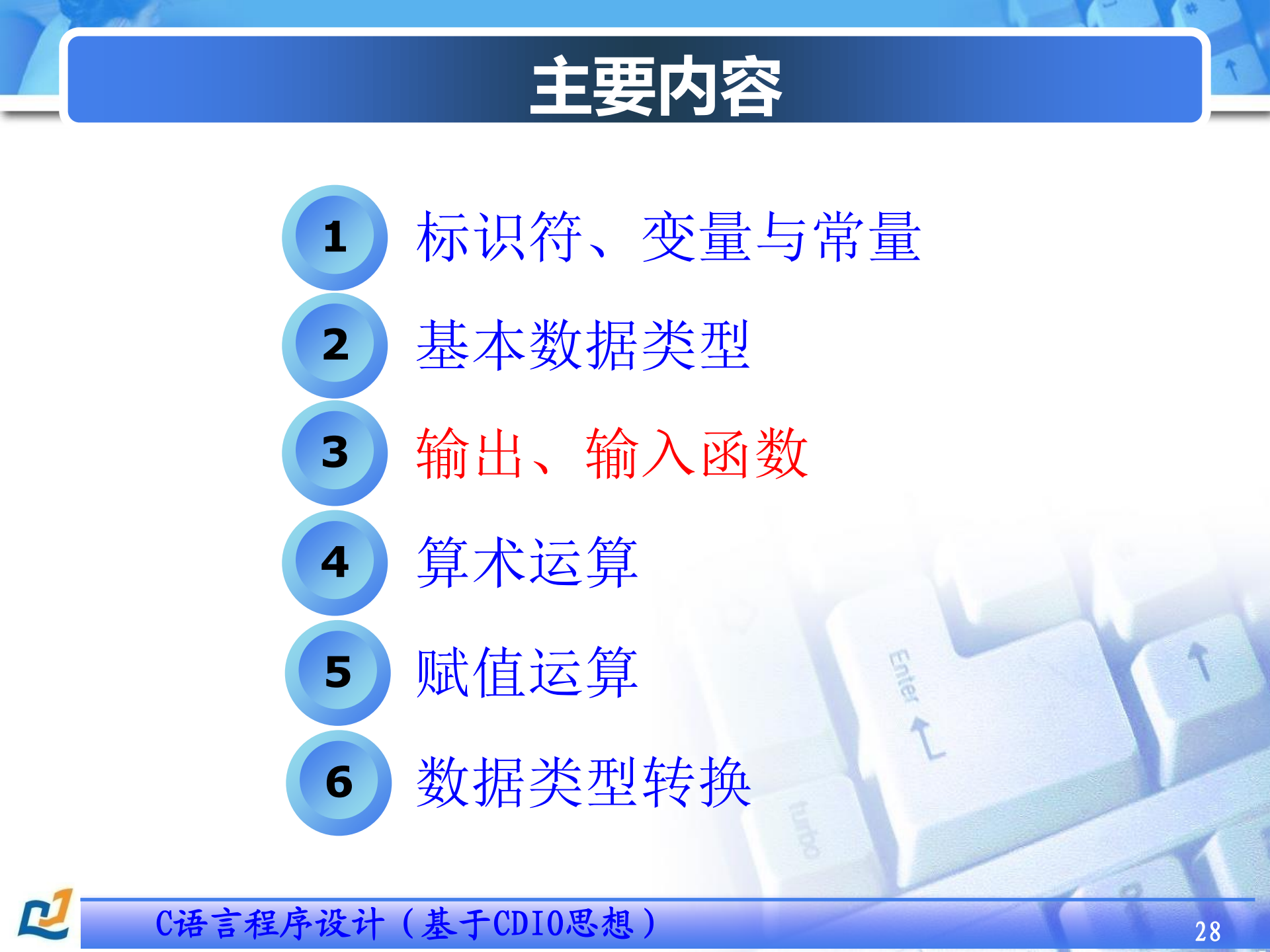
“a” | 97 | \0 |



- 是ASCII值为0的字符NULL
- 是空操作符
- 是非显示字符

主要内容

1. 标识符、变量与常量
2. 基本数据类型
3. 输出、输入函数
4. 算术运算
5. 赋值运算
6. 数据类型转换



格式输出函数printf()

- 格式输出函数:

  printf(“格式说明”，输出项目表)

  - 功能：按“格式说明” 的输出格式，将输出项目表中的各输出项依次输出到系统指定的缺省输出设备。
  - 输出项目表是用逗号分隔的0～n个表达式
  - “格式说明”由“%”和格式字符组成，格式字符如表中所示

格式输出函数printf()

格式输出函数:

格式字符	输出项形式
d，I	十进制整数（正数不输出符号）
x，X	无符号十六进制整数,用大写 X 时，输出十六进制的大写字母(A…F)
O	无符号八进制整数
U	无符号十进制整数
F	以小数形式输出实数，隐含输出六位小数
e，E	以指数形式输出实数，用大写 E 时，指数用大写 E 表示
g,G	选用%f 或%e 格式中输出宽度较短的一种格式，用大写 G 时，指数用大写表示
C	输出一个字符
S	字符串

格式输出函数printf()

- 附加格式字符可加在格式字符d、o、x、u前
  - m：一个正整数，表示数据的最小宽度
  - .n：一个正整数，表示小数点后面的位数

输出举例：

- a=123;b=12345;

  printf(“a=%4d,b=%4d”,a,b);

  输出：a= 123,b=12345

- j=586;y=3.1415;

  printf(“%5d\n”,j);　　输出：　586

  printf(“%-5d\n”,j);　　586

  printf(“%7.2f\n”,y);　　3.14

格式输出函数printf()

例：格式说明符的宽度和精度使用说明

```
#include "stdio.h"
main()
{ float a=12345.678;
  int b=12345;
  printf("\n12345678901234567890");
  printf("\n%21.10f:a1",a);
  printf("\n%2.2f:a2",a);
  printf("\n%10d:b1",b);
  printf("\n%2d:b2",b);
}
```

程序运行结果：

```
12345678901234567890
     12345.6777343750:a1
12345.68:a2
      12345:b1
12345:b2
```

格式输出函数printf()

例：将上例结果按左对齐输出

```
#include "stdio.h"
main()
{ float a=12345.678;
  int b=12345;
  printf("\n12456789012345678 90");
  printf("\n%-21.10f:a1",a);
  printf("\n%-2.2f:a2",a);
  printf("\n%-10d:b1",b);
  printf("\n%-2d:b2",b);
}
```

程序运行结果：

```
12456789012345678 90
12345.6777343750     :a1
12345.68:a2
12345      :b1
12345:b2
```

格式输出函数printf()

例：数据前用0填补示例

```
#include "stdio.h"
main()
{ float a=12345.678;
  int b=12345;
  printf("\n12345678901234567890");
  printf("\n%021.10f:a1",a);
  printf("\n%02.2f:a2",a);
  printf("\n%010d:b1",b);
  printf("\n%02d:b2",b);
}
```

程序运行结果：

```
12345678901234567890
000012345.6777343750:a1
12345.68:a2
0000012345:b1
12345:b2
```

格式输入函数scanf()

格式输入函数：

scanf(“格式说明”，地址表列)

- **功能：从键盘上输入数据，并按照指定的输入格式把数据赋给相应的输入项**
- **“格式说明”由“%”和格式字符组成，如%c,%d**
- **地址表列由变量的地址组成，如：&a,&b，&是取地址运算符**
- **实例：**

```
scanf(“%d%d”,&a,&b);
scanf(“%d,%d”,&a,&b);
scanf(“a=%d,b=%d”,&a,&b);
```

输入：3 4

输入：3,4

输入：a=3,b=4

scanf(“%d”,x); scanf(“%f”,x+6);? 错误用法

输入/输出举例

```
main()
{char ch;
 int i;
 float x;
 scanf(“%c %d %f”,&ch,&i,&x);
 printf(“ch=%c i=%d x=%f\n”,ch,i,x);
}
```

输入：d 23 12.345

输出：ch=d i=23 x=12.345000

输入：d 23.645 12.345

输出：ch=d i=23 x=0.645000

- 输入数据时说明符间的字符也应该输入
- 输入的数据类型必须与格式说明的一致
- 输入数据的个数不能少于格式说明的个数

输入/输出举例

编程:计算从键盘输入的一个数的平方值和立方值，并显示出来。

```
main()
{int x,y,z;
 printf("\nx=");
 scanf("%d",&x);
 y=x*x; z=x*x*x;
 printf("输入值=%d\n",x);
 printf("平方值=%d\n",y);
 printf("立方值=%d\n",z);
}
```

输入：x=5

输出：输入值=5

平方值=25

立方值=125

输入/输出举例

编程:从键盘输入某电视机的价格，再输出该价格打7折后的价格。

```
main()
{float  x,y;
 printf(“\nx=”);
 scanf(“%d”,&x);
 y=x*0.7;
 printf(“输入价格=%d\n”,x);
 printf(“打7折后的价格:%d\n”,y);  }
```

输入：x=100

输出：输入价格=100

打7折后的价格：70

输入/输出举例

- 编程：求一元一次方程**ax+b=0**的根。

  分析：方程的根即**x**的值为：**-b/a**

```
main()
{float  a,b,x;
 printf("\na,b=");
 scanf("%f,%f",&a,&b);
 x=-b/a;
 printf("a,b=%.2f,%.2f\n",a,b);
 printf("方程的根为:%.2f\n",x);
}
```

输入：

a,b=3,6

输出：

a,b=3.00,6.00

方程的根为:-2.00

缺少必要安全性检查，程序不完整。

主要内容

1. 标识符、变量与常量
2. 基本数据类型
3. 输出、输入函数
4. 算术运算
5. 赋值运算
6. 数据类型转换

运算符及表达式

- 运算符是表示某种运算的符号，是对数据的操作
- 表达式是用运算符和括号将运算对象连接起来的符合C语言语法规则的式子
- 运算对象可以是常量、变量、函数等
- 优先级和结合性:优先级是指表达式中各计算的先后次序；结合性是指当一个运算对象两侧的运算符的优先级相同时进行运算的结合方向

运算符及表达式

优先级	运算符	结合方向
1	() [] .(取成员)—>(指向成员) ++ --（后缀）	→
2	!（逻辑非）~（按位取反） ++ --（前缀）-(负) *(间接引用) &(取地址) sizeof （容量运算）	←（右结合）
3	* / %	→
4	+ -	→
5	<< （左移运算） >>（右移运算）	→
6	< <= > >= （关系运算）	→
7	== != （关系运算）	→
8	&（按位与）	→
9	^（按位异或与）	→
10	|（按位或）	→
11	&& （逻辑与）	→
12	|| （逻辑或）	→
13	? : （条件运算）	←（右结合）
14	= += -= *= /= %= <<= >>= &= ^= |= （赋值运算）	←（右结合）
15	,（逗号运算）	→

算术运算符

- 算术运算符有：
  - 单目运算符：-（负）、+（正），右结合
  - 双目运算符：+、- 、*、/(整数相除结果取整)、%（取余）
- 优先级：先乘除(含取余),后加减,括号优先
- 结合性：同级从左至右
- 写出以下面算术表达式的优先级和结合性：

  a*b/c-1.5+’a’

  a/b/c*(3-d)

要求%两侧均为整形数据

算术表达式

- 用算术运算符和括号将运算对象连接起来的、符合C语法规则的式子。例如：
  - 25/4 等于 6， 25.0/4.0 等于 6.25
  - 5/10 等于 0， 5.0/10.0 等于 0.5
  - 4%9=4 -15%4=-3(%只能对整型数据操作)
- 乘号不能省略，且要根据运算顺序书写，如：

数学表达式	C表达式
$a(b^2+4ac)$	a*(b*b+4*a*c)
$(a+b)\div cd$	(a+b)/c/d 或 (a+b)/(c*d)

数据类型与运算结果的关系

- 同类型数据的运算结果仍保持原数据类型。
  - 整型数的除法得到的结果仍是整型数，小数部分会被去掉。浮点数除法得到的仍是浮点数。

    如：25/4 等于 6， 25.0/4.0 等于 6.25

    5/10 等于 0， 5.0/10.0 等于 0.5

    4%9=4 -15%4=-3（取模运算%，只能对整型数据操作）

- 不同数据类型混合运算，低类型往高类型转换后再做运算，以保持运算结果不损失精度。

  如：5.0/2 = 2.5

算术运算举例

```
main()
{int i=3;
 float r=2.0;
 printf("2*-i:%d\n",2*-i);
 printf("r/i:%f\n",r/i);
 printf("r/i:%d\n",r/i);
 i=r/i;
 printf("i=r/i:%d\n",i);
 i=2%3;
 printf("2%%3:%d",i);
}
```

程序执行结果:

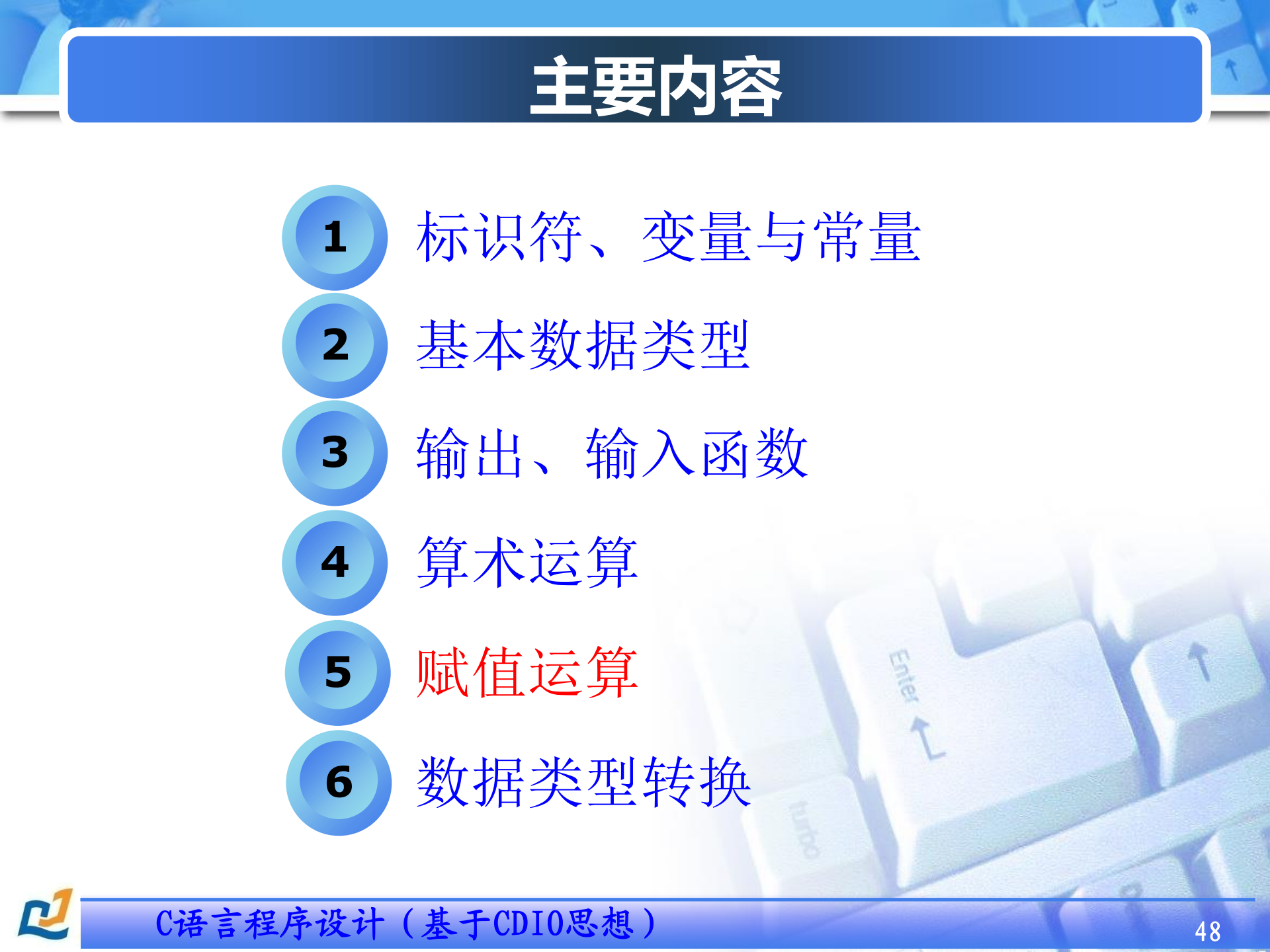
```
2*-i:-6
r/i:0.666667
r/i:1610612736

i=r/i:0

2%3:2
```

算数运算实战演练

- **1.编程序计算数学表达式：b^2-4ac,a,b,c的值从键盘输入。**
- **2.编程序计算298秒是几分几秒。**
  - 提示：设 int x=298; 再定义两个变量存放分（m）、秒(s)值; 则：m=x/60;s=x%60;
- **3.从键盘输入一个三位数，求各位数字之和。例如，输入的三位数为358，则输出结果为3+5+8=16 。**
  - 提示： 题目的关键是要求出该数的个、十、百位上的数字，可利用C语言整数相除，结果仍为整数的特点。若设该数为data,它的个、十、百位为g、s、b，则b=data/100 ;s=(data-b*100)/10;g=data%10

主要内容

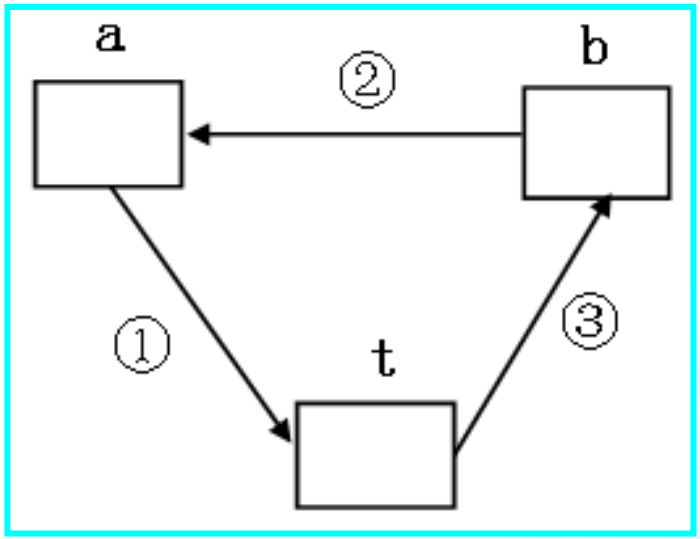
1. 标识符、变量与常量
2. 基本数据类型
3. 输出、输入函数
4. 算术运算
5. 赋值运算
6. 数据类型转换

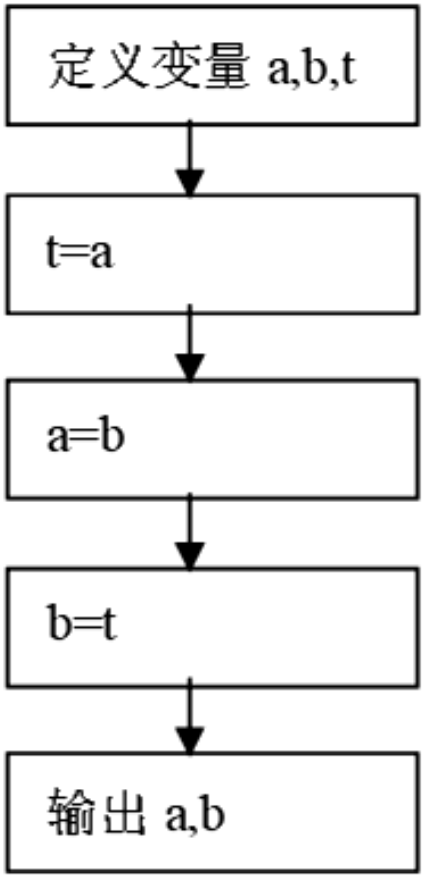


赋值运算

例：交换a,b变量的值

```
main()
{int a=3,b=5,t;

 t=a;
 a=b;
 b=t;

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





赋值运算符及表达式

- 由赋值运算符“=”将变量和表达式连接起来的式子。

<变量>=<表达式>

赋值表达式	表达式值	各变量值
a=5	5	
a=b=c=5	5	a, b, c均为5
a=5+(c=6)	11	a为11，c为6
a=(b=4)+(c=6)	10	a为10，b为4，c为6
a=(b=10)/(c=2)	5	a为5，b为10，c为2

- 赋值语句的格式：赋值表达式;
- 赋值表达式可以包含在其它表达式中，而赋值语句不能，如：

```
if((a=b)>0) t=a;
```

复合赋值运算符

- C语言中有10种**复合赋值运算符**：

  +=、 -=、 *=、 /=、 %=、

  <<=、>>=、&=、^=、|= （位运算符不在考试要求范围内）

- **实例：**

a/=3	等价于	a=a/3
x*=y+8	等价于	x=x*(y+8)
a+=a-=a*a	相当于	a=a+(a=a-a*a)
y%=8	等价于	y=y%8

复合赋值运算符

例：　复合赋值运算符的应用

```
main()
{ int x,y;
  x=3; y=8;x*=y+1;
  printf("x=%d,y=%d\n",x,y);
  x=3;y=8;x=x*y+1;
  printf("x=%d,y=%d\n",x,y);
}
```

程序执行结果：

```
x=27,y=8
x=25,y=8
```

自增、自减运算

- ++(自增)，　--(自减)：
- ++n，--m　　前缀，变量的值+(-)1
- n++ ，m--　后缀，　　变量的值+(-)1
- 优先级：高于双目运算
- 结合性：同级从右至左

说明：对于j变量，j++和++j都使j的值加1，但k的值就不同了，所以前缀和后缀运算对变量是一样的，对表达式的值就不一样

```
j=3;
k=++j;  → j=j+1;k=j;   即j的为4，k的值为4
k=j++;  → k=j;j=j+1;   即j的为4，k的值为3
```

自增、自减运算举例

例：自增1运算,前后缀区别

```
#include "stdio.h"
main()
{ int x,y;
  x=5; y=x++;
printf("x=5,y=x++:%d,x=%d\n",y,x);
  x=5;y=++x;
printf("x=5,y=++x:%d,x=%d\n",y,x);
}
```

运行结果:

x=5, y=x++: 5, x=6

x=5, y=++x: 6, x=6

赋值运算实战演练

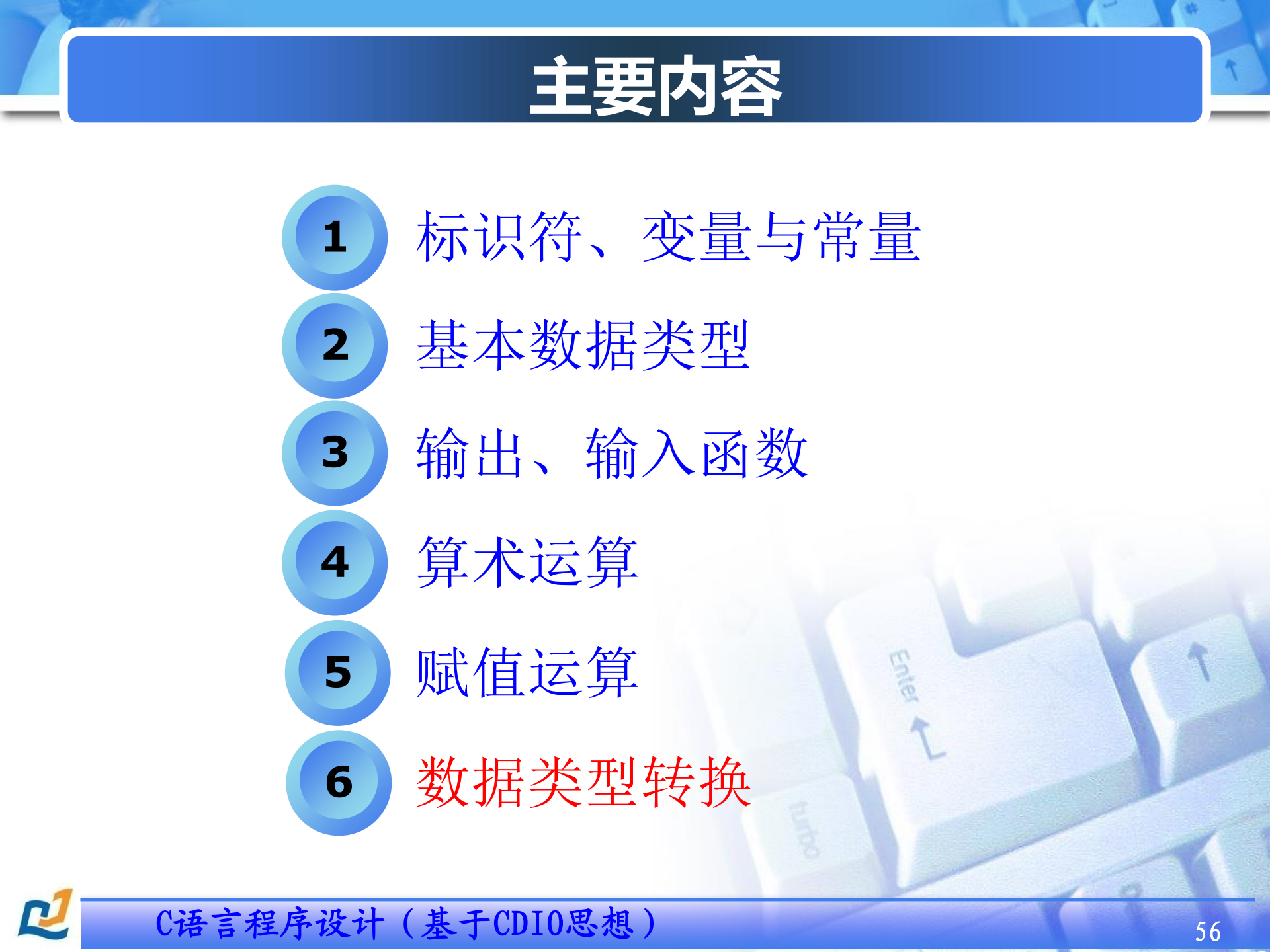
1.用赋值语句表达：a 的值为2，b的值比a多2,c 的值比b的值多2，并输出a,b,c的结果。请编出完整程序验证。

2.假设变量a和b均为整型，以下语句可以不借助任何变量把a、b中的值进行交换。请先填空，再编出完整程序验证。

a+=_____;(a为a、b之和) b=a-_____;a-=_____;

主要内容

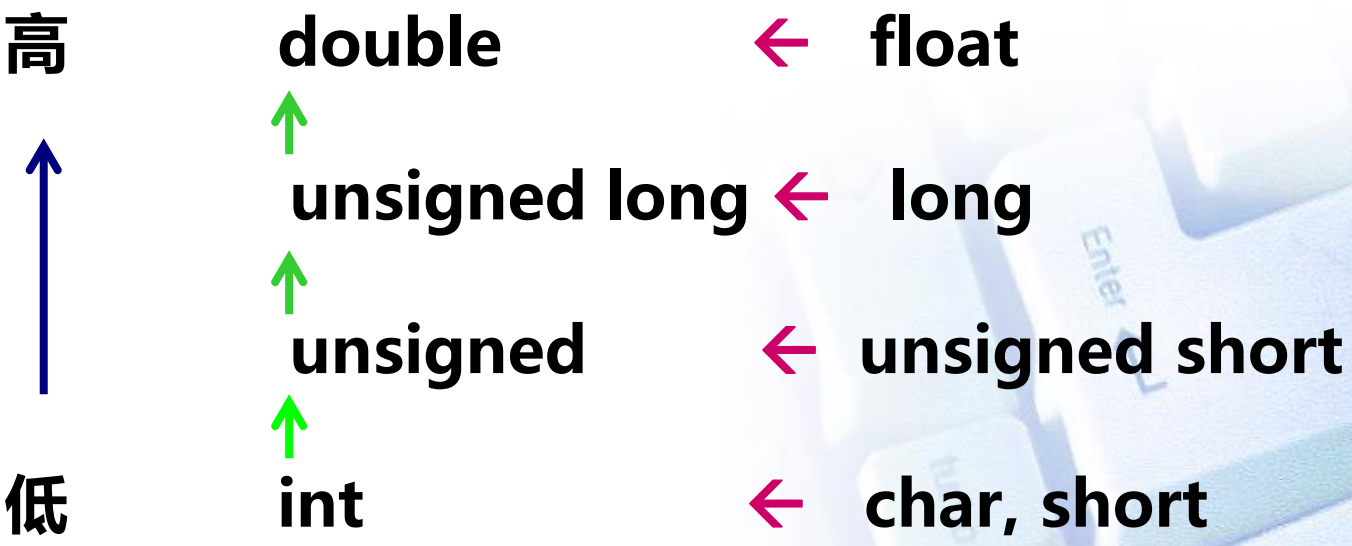
1. 标识符、变量与常量
2. 基本数据类型
3. 输出、输入函数
4. 算术运算
5. 赋值运算
6. 数据类型转换



数据类型转换

自动转换（隐式转换）：当参加算术运算的数据类型不一致时低级向高级转换；赋值运算符两边的数据类型不同时，将右侧表达式的值转换为左侧变量的类型

- 水平方向：自动
- 垂直方向：低 → 高



自动转换举例

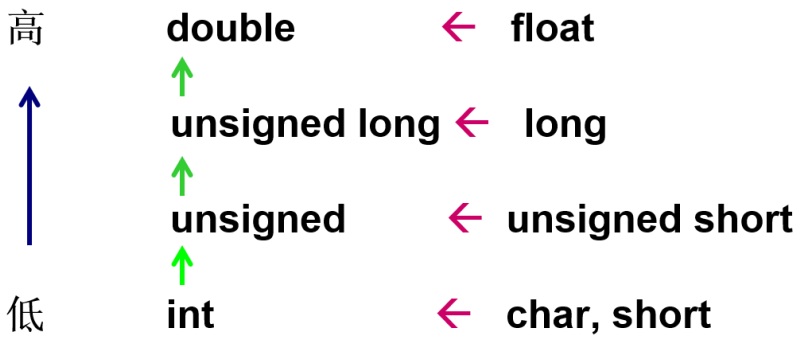
自动转换（隐式转换）

'A' + 12 – 10.05

65

77

66.95



Try: `f` is defined to be a `float`, `i` an `int`, `l` a `long int`, and `s` a `short int` variable,

evaluate expression `f * i + l / s`

What's the type of the evaluation result?

数据类型转换

强制转换（显式转换）：强制变量或表达式的值转换为某一特定类型。常用在自动类型转换不能达到目的时。

转换格式为:

(类型说明符) 变量

(类型说明符)(表达式)

*转换不会改变变量定义时所规定的数据类型

(double)a，　(int)(x+y) ，(float)(5%3)

(double)3	3.0
(int)3.8	3
(double)(5/2)	2.0
(double)5/2	2.5

强制转换举例

强制转换（显式转换）

```
# include <stdio.h>
int main(void)
{
     int i;
     double x;
    x = 3.8;
    i = (int) x;
    printf("x = %f, i = %d \n", x, i);
    printf("(double)(int)x = %f\n", (double)(int)x);
    printf(" x mod 3 = %d\n", (int)x % 3);
    return 0;
}
```

x = 3.800000, i = 3
(double)(int)x = 3.000000
x mod 3 = 0

实战演练

- 设银行定期存款的年利率r为2.25%，并已知存款期为n年，存款本金为m元，编程计算n年后的本利之和

1、分析与设计

根据题意算法步骤为：

step 1：定义变量；

step 2:输入算法所需要的数据r,n,m；

step 3：进行运算和数据处理:本利之和=$m*(1+r)^n$

step 4：输出运算结果数据。

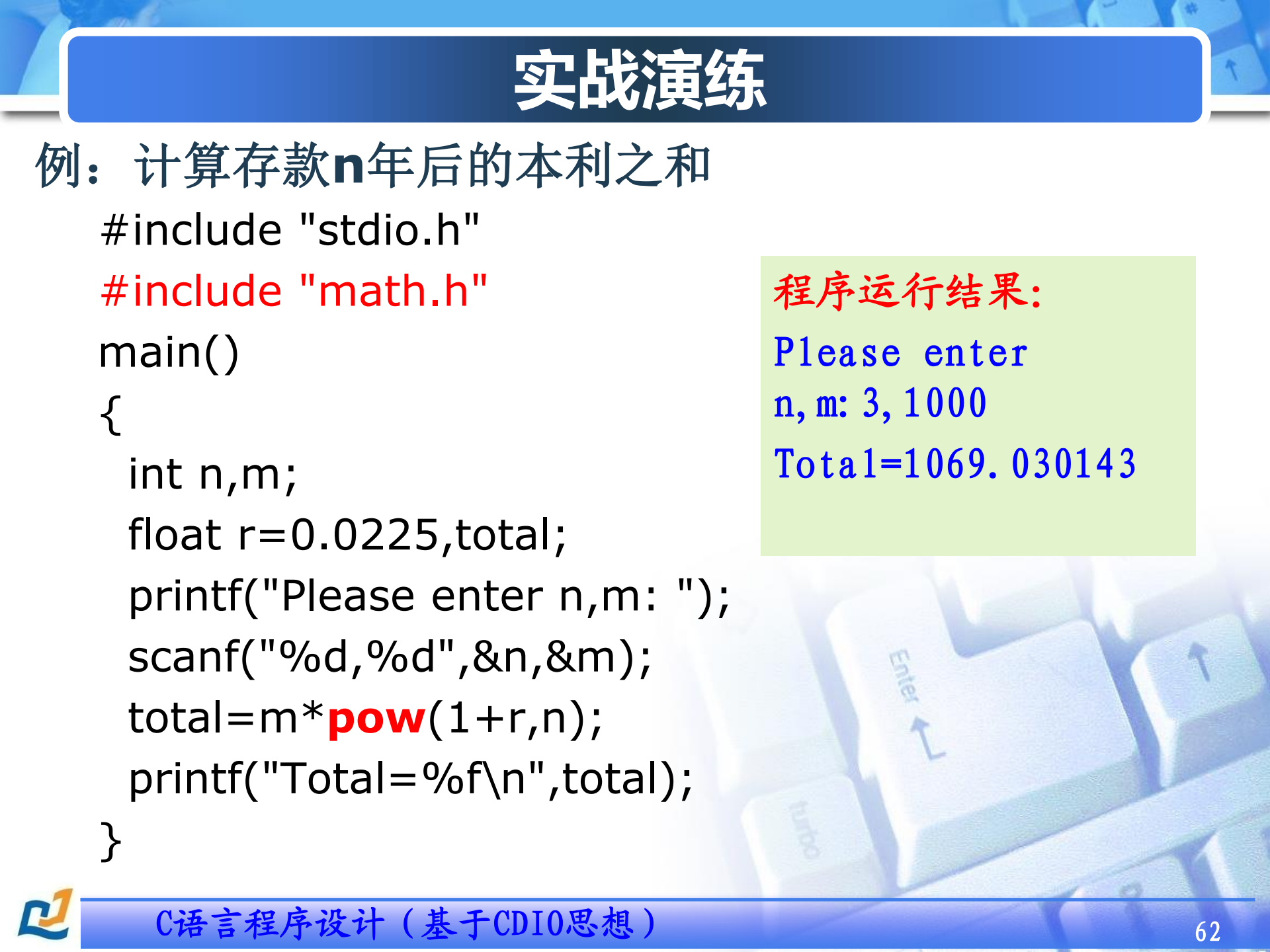
实战演练

例：计算存款n年后的本利之和

```
#include "stdio.h"
#include "math.h"
main()
{
  int n,m;
  float r=0.0225,total;
  printf("Please enter n,m: ");
  scanf("%d,%d",&n,&m);
  total=m*pow(1+r,n);
  printf("Total=%f\n",total);
}
```

程序运行结果:

```
Please enter
n,m: 3,1000
Total=1069.030143
```



Thank you