



清华大学

腾讯会议 ID: 507 639 672
Tsinghua University

计算机程序设计基础

第2讲 顺序程序设计与 基本数据类型（一）

沈瑜 (010-62782951)

shenyu@tsinghua.edu.cn

清华大学电机系

2022.9.20



第一次上机课之后，对编程环境熟悉得如何了？

- A 感觉很好，基本掌握编程环境
- B 有少量不明白之处，有信心能跟上！
- C 完全懵了，需要抓紧学习！
- D 由于各种原因，还没尝试上机（补选课等）

提交





主要内容

- 简单的C程序设计举例
 - 顺序程序设计，实现简单的数学计算
- 简单的数据类型（一）整数与实数
- 运算符与表达式
- 简单数据的输入输出（一） **printf**

参考教材：《谭浩强》第3章；《现代方法》第2,3,4,7章
（算法有关的内容先不细看/字符有关的下一节课讲）
网络学堂：“VS2012安装及初级使用指南”





2.1 简单的C程序设计举例

●例1：求两个整数之和。

```
#include <stdio.h>    //预编译处理命令

//求两个整数之和
int main()             //主函数
{                       //开始
    int a, b, sum;      //声明部分，定义3个整型变量

    a = 123;            //对变量a赋值
    b = 456;            //对变量b赋值
    sum = a+b;          //进行加法运算，结果存放在变量sum中

    printf( "sum is %d \n", sum ); //输出两数之和

    return 0;           //函数 返回值为0
}                       //函数结束
```



●例2: 求两个实数之和。

```
#include <stdio.h> //预编译处理命令

//求两个实数之和
int main()           //主函数
{                   //函数开始
    float a, b, sum; //声明部分, 定义3个浮点型变量

    a = 123.456;      //对变量a赋值
    b = 456.789;      //对变量b赋值
    sum = a+b;        //进行加法运算, 结果存放在变量sum中

    printf( "sum is %f \n", sum ); //输出两数之和

    return 0;         //函数 返回值为0
}                     //函数结束
```

●例3: 计算圆的面积。

```
#include <stdio.h> //预编译处理命令

#define PI 3.1415926535897932384626
//求圆的面积

int main() //主函数
{ //函数开始
    float r, area; //声明部分，定义2个浮点型变量

    r = 30.5; //对半径变量r赋值
    area = PI * r * r; //进行面积计算，结果存放在变量area中

    printf( "PI is %f \n", PI ); //输出PI值
    printf( " r is %f, so area is %f \n", r, area );
    //输出半径和面积

    return 0; //PI is 3.141593
} // r is 30.500000, so area is 2922.466553
```



2.2 简单的数据类型（一）整数与实数

1. 常量 **constant**

- 整数 **int**

2 -23 2456 0

- 浮点数（实数） **float**

十进制小数形式 2.75

指数形式 3.16E3 （代表 3.16×10^3 ）

-2e-8 （代表 -2×10^{-8} ）

上述数据，称为常量。 **Constant**

符号常量:

```
#define PI 3.1415926535897932384626
```

*整数(int)的计算机表达形式

人类计数方式：“逢十进一”

$$178 = 1 \times 10^2 + 7 \times 10^1 + 8 \times 10^0$$

计算机：

二进制 (Binary)

$$13 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (1101)_2$$

bit (位) 比特

Byte / 'bait/ (字节) = 8 bit

八进制 (Octal)

(262)₈ 0262

十六进制 (Hexadecimal)

(B2)₁₆ 0xB2



*整数的表值范围、负整数的表示

正整数的表值范围:

对**1Byte**即**8位**, 表示正整数的范围为:

0 ~ 0xff 即 **0 ~ 255**

对**2Byte**即**16位**, 表示正整数的范围为:

0 ~ 0xffff 即 **0 ~ 65535**

只有加法器的计算机:

溢出现象 $(11111111)_2 + 1 \equiv 0$ (相当于取模运算)

减法? 乘法? 除法?

负整数的表示:

1Byte **-128 ~ 127**

$(11111111)_2 + 1 \equiv 0 \rightarrow (11111111)_2$ 即为 **-1**

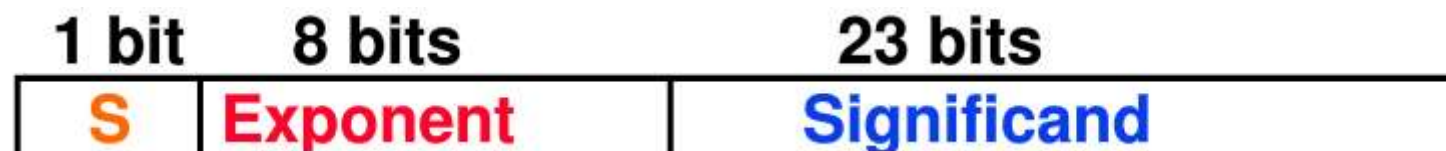
最高位为符号位



*浮点数(float)的计算机表达形式

✚ 计算机中以指数形式存放实数

※ 小数部分（含符号）、指数部分



$$(-1)^S \times (1 + \text{Significand}) \times 2^{(\text{Exponent}-127)}$$

※ float: 4个字节，含6~7位有效数字

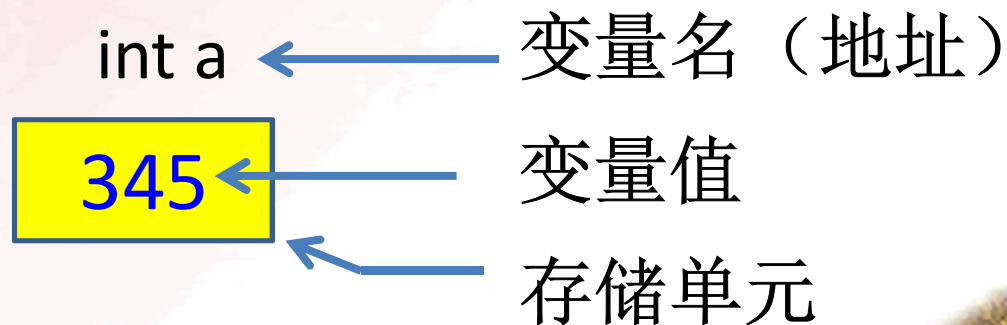
※ double: 8个字节，含15位有效数字

注意：尽量不要尝试直接比较两个实数是否相等，小于某一个精度则认为相等。



2. 变量 variable

- 变量，用以临时存储数据
- 变量必须先**定义**，后**使用** `int age; float height;`
- 定义变量时指定该变量的**名字**和**类型**
- **变量名**和**变量值**是两个不同的概念
- 变量名实际上是以一个名字代表的一个**存储地址**
- 从变量中取值，实际上是通过变量名找到相应的内存地址，从该存储单元中读取数据



变量类型

- 整型
 - short int, int, long int, long long
 - unsigned int, unsigned long
- 实型
 - float, double, long double
- 数值的表达范围大小不同；内存字节数不同
 - 与编译器/操作系统有关
- **sizeof** 关键字 **不是函数**
sizeof(int) sizeof(intA)
取得字节数。编译时处理，相当于一个常量



3. 标识符 identifier

- 变量、函数、宏、对象命名，名字是标识符

大小写字母是不同的字符

case sensitive

- C 语言规定标识符只能由**字母**、**数字**和**下划线**3种字符组成，且**第一个字符必须为字母或下划线**
- 合法的标识符：如**sum, average, _total, Class, day, BASIC, li_ling, times100**
- 不合法的标识符：**M.D.John, ¥123, #33, 3D64, a>b, li-ling**
- 推荐“驼峰规则”：不用 **int studentnumber**
用 **int nStudentNumber**



4. 变量的声明和赋值

- 变量使用前必须先声明

- 早期标准

```
int main()
{
    int age, num;
    float height;
    age=18;
    height=1.81;
    return 0;
}
```

- = 称为赋值运算符

- C99标准 及 C++

```
int main()
{
    int age, num;
    age=18;
    float height;
    height=1.81;
    return 0;
}
```



5. 变量的初始化

- 程序开始执行时，有一些变量自动设置为零，但是很多变量的值是不定的（**未初始化**）。
 - 若使用未初始化的变量，结果不可预知/甚至程序崩溃
- 除了赋值的方法给变量赋初始值，还可初始化：
float height = 1.81;
1.81称为 初始化式 (initializer)
- 这里的 **= 初始化** 与 **赋值运算符 =** 不是一回事！
- 以下两种初始化写法：
float height=1.81, length=6.6, width= 3.1;
float height, length, width= 3.1;





2.3 运算符与表达式

1. 基本的算术运算符

- + : 正号运算符(单目运算符) Unary Operator
- : 负号运算符(单目运算符)
- + : 加法运算符
- : 减法运算符
- * : 乘法运算符
- / : 除法运算符
- % : 求余运算符



算术运算的说明

- 两个整数相除的结果为整数
 - 如**5/3**的结果值为 1，舍去小数部分
 - 如果除数或被除数中有一个为负值，舍入方向不固定。例如，**-5/3**，有的系统中得到的结果为**-1**，在有的系统中则得到结果为**-2**
 - **VC++**采取“向零取整”的方法
如**5/3=1**，**-5/3=-1**，取整后向零靠拢
- **% 运算符**要求参加运算的运算对象(即操作数)为整数，结果也是整数。结果的符号与被除数符号相同。如**11%3**，结果为**2**；**-5%3**，结果为**-2**



2. 自增、自减运算符

- 作用是使变量的值 1 或减 1
 - `++i;` 在使用*i*之前, 先使*i*的值加1
 - `--i;` 在使用*i*之前, 先使*i*的值减1
 - `i++;` 在使用*i*之后, 使*i*的值加1
 - `i--;` 在使用*i*之后, 使*i*的值减1
- 对于复杂的表达式中, 慎用
尤其是复杂的优先级和结合顺序时慎用



有以下代码语句：

```
int i=2, j=3, m, n, k;  
m=(i++)*(++j);  
n=(++i)*(j++);  
k=(++i)*(++i);
```

运行上述代码语句后，m,n,k的值是多少？

m = [填空1]

n = [填空2]

k = [填空3]

作答

慎用



3.算术表达式和运算符的优先级与结合性

- 用算术运算符和括号将运算对象（也称操作数）连接起来的、符合C语法规则的式子，称为C算术表达式
- 运算对象包括常量、变量、函数等
- C语言规定了运算符的优先级和结合性
 - 圆括号的优先级最高
 - 教材附录D“运算符和结合性”， p 378



4.不同类型数据间的混合运算

- **+**、**-**、*****、**/** 运算的两个数中有一个数为**float**或**double**型，结果是**double**型。系统将**float**型数据都先转换为**double**型，然后进行运算
- 如果**int**型与**float**或**double**型数据进行运算，先把**int**型和**float**型数据转换为**double**型，然后进行运算，结果是**double**型



设x,y均为int型变量, 且 $x=1$, $y=2$,则表达式 $1.0+x/y$ 的值为

- ☐ A 0
- ☐ B 1
- ☒ C 1.0
- ☐ D 1.5

提交



5. 强制类型转换运算符

- 强制类型转换运算符的一般形式为

(类型名) (表达式)

(double)a (将 a 转换成**double**类型)

(int) (x+y) (将**x+y**的值转换成**int**型)

(float)(5/3) (将**5/3**的值转换成**float**型)

- 有两种类型转换
 - ◆ 系统自动进行的类型转换
 - ◆ 强制类型转换



6. 赋值过程中的类型自动转换

- 如果赋值运算符两侧的类型一致，直接赋值

`i=234;` // 此前*i*已经被定义为int型

- 如果赋值运算符两侧的类型不一致，自动转换

- ◆ 将浮点数赋给整型变量时，**取整**，舍弃小数部分

`i = 3.45;` //结果是 `i=3`

- ◆ 整型数赋给浮点数，数值不变，浮点数形式存储

`f = 23;` //此前*f*被定义为float或double，相当于*f*=23.0

- ◆ 字符型数据赋给整型变量，赋给ASCII码

`i = 'A';` //此前*i*已经被定义为int型

- ◆ **截断**现象（给允许表值范围小的变量赋值时常见）

`c=289;` //c 已被定义为char型，实际*c*=33



7*. 逗号表达式与典型表达式示例

- 逗号运算符（又称“顺序求值运算符”）

表达式1, 表达式2

逗号表达式的求解过程是：先求解表达式1，再求解表达式2。
整个逗号表达式的值是表达式2的值。

3+5, 6+8 //表达式的值为14

- 示例：

C=5*(100-32)/9; //? 华氏100度转换; 37.000000

C=5/9*(100-32); //? 华氏100度转换; 0.000000

a=b=1; //赋值运算符，自右向左

a=3*5, a*4; //表达式值60; 赋值运算符优先级高于逗号运算符

x=(a=3, 6*3); //表达式值18

x=a=3, 6*a; //表达式值18





2.4 简单数据的输入输出（一）

1. 输入输出的概念

- 以计算机主机为主体而言
 - 计算机向输出设备（显示器/打印机）输出
 - 计算机从输入设备（键盘/磁盘/扫描仪等）输入
- C标准函数库的输入输出函数

printf **scanf**

#include <stdio.h>

这两个函数是格式输入输出函数
用这两个函数时，可指定格式



2. printf显示变量的值/表达式的值

- 程序在初始化所有变量之后，就找到**main**函数入口，逐个语句执行；
- 用**printf**函数可显示变量的**当前值**

```
int age = 18;
```

```
printf("age: %d\n", age);
```

```
float height = 1.81; //浮点数
```

```
printf("height: %f\n", height );
```

```
height = 20.7;
```

```
printf("height: %f\n", height );
```

- **占位符****%d**和**%f**用以指明变量的值的显示位置



printf显示表达式的值

- 用printf函数可显示表达式的值

```
float height=1.81, length=6.6, width= 3.1;  
printf("volume: %f\n", height*length*width);
```

```
int age=18;
```

```
printf("age: %d\nheight: %f\nlength: %f\nwidth:  
%f\nvolume: %f\n", age, height, length, width,  
height*length*width);
```

- 显示多个变量/表达式的值
- 占位符%d和%f，与对应变量类型必须一致，否则结果难以预料



3. printf函数的用法

printf函数的一般格式

printf（格式控制，输出表列）

例如：

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

格式声明

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

普通字符

可以输出常量、
变量或表达式



printf函数的常用格式字符

- **d** 格式符。用来输出一个有符号的十进制整数
 - 可以在格式声明中指定输出数据的域宽

```
printf("%5d%5d\n",12,-345);
```

 - **%d**输出int型数据
 - **%ld**输出long型数据
- **f**格式符。用来输出实数，以小数形式输出
 - ①不指定数据宽度和小数位数，用**%f**

decimal

float

例：用**%f**输出实数，只能得到 6 位小数。

```
double a=1.0;
```

```
printf("%f\n",a/3);
```

```
0.333333
```



printf函数的常用格式字符

- f格式符

① 不指定数据宽度和小数位数，用%f

② 指定数据宽度和小数位数。用%m.nf

```
printf("%20.15f\n",1/3);
```

```
0.333333333333333
```

```
printf("%.0f\n",10000/3.0);
```

```
3333
```

③ 输出的数据向左对齐，用%-m.nf



printf函数的常用格式字符

- **f**格式符
 - 用来输出实数，以小数形式输出
 - **float**型数据只能保证**6**位有效数字
 - **double**型数据能保证**15**位有效数字
 - 计算机输出的数字不都是绝对精确有效的



printf函数的常用格式字符

- **e**格式符。指定以指数形式输出实数

- **%e**，VC++给出小数位数为6位

指数部分占**5**列

小数点前必须有而且只有**1**位非零数字

```
printf("%e",123.456);
```

输出: 1.234560 e+002

- **%m.ne**

```
printf("%13.2e",123.456);
```

输出: **1.23e+002** (前面有4个空格)



例：方程求解

求 $ax^2 + bx + c = 0$ 方程的根。**a、b、c**暂时固定。

设 $b^2 - 4ac > 0$

如果 $b^2 - 4ac \geq 0$ ，则一元二次方程有两个实根：

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

若记 $p = \frac{-b}{2a} \quad q = \frac{\sqrt{b^2 - 4ac}}{2a}$

$$x_1 = p + q$$
$$x_2 = p - q$$




```
#include <stdio.h>
#include <math.h>
int main ( )
{ double a,b,c,disc,x1,x2,p,q;
  a=1.0, b=1, c = -30; //后续改为键盘输入
  disc=b*b-4*a*c;
  p=-b/(2.0*a);
  q=sqrt(disc)/(2.0*a);
  x1=p+q; x2=p-q;
  printf("x1=%7.2f\nx2=%7.2f\n",x1,x2);
  return 0;
}
```

输出数据占7列，其中小数占2列