# 第二节课

整型浮点字符

**运算输入输出**

1. **运算符与表达式**

运算符：动作

算子：数据（）

运算符分类：

1. 算数运算符：目（算子数量）；优先级（乘法）；结合律



定义int后‘/’是整除

‘/数字.’加了“.”会发生类型转换

注意边界 注意表示范围

1. 自增自减运算符

++ --前置后置不同

前置表示：先对变量加（减），再赋值

后置：先赋值，再加（减）

1. 赋值运算符和赋值表达式

赋值只能赋变量

常量、const 变量、表达式、函数调用均不能作为左值，而右值可以是常量、变量、函数调用或表达式。

如果赋值运算符两侧的类型不一致，但都是数值型或字符型时，在赋值时要进行类型转换。转换过程中可能会产生 精度丢失、数据错误等复合赋值运算符

如：

x += 2 等价于 x = x + 2

x -= 2 等价于 x = x – 2

x \*= 2 等价于 x = x \* 2

x /= 2 等价于 x = x / 2

x %= 2 等价于 x = x % 2

x \*= y + 8 等价于 x = x \* (y + 8)（算数比赋值优先级高）

从右往左算

嵌入赋值

int a, b, c, m=25;

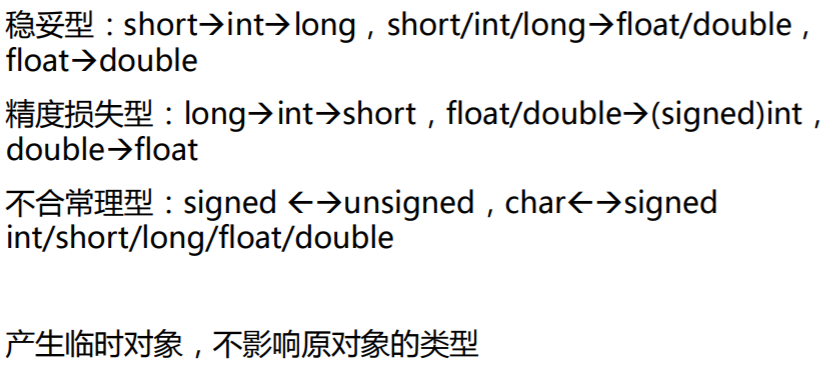
c = (a=12) % (b=5);

1. 类型转换运算符

格式：（类型名）（表达式）

但是只是临时对象，运行完之后还是初始定义

隐式类型转换



1. 位运算符及表达式（了解，我并不知道有啥用）

对象：无符号整型

左移与右移

1. 取长度运算符
2. 逗号运算符（级别最低，返回值是右边的表达式）

Printf (“%d, %d, %d”, a, b, c);

Printf (“%d, %d, %d”,(a ,b, c),b, c) （这个里面逗号就是运算符）

**该加括号加括号**

1. **C语言语句**

表达式语句 复合语句 函数调用语句

控制语句：条件判断

循环执行语句

转向语句

空语句：只有一个分号的语句

作用：延时 条件分支配对 循环体 定义空函数

1. **数据输入输出**

#include<stdio.h>

Input 非缓冲型（马上有回应）：getch ()屏幕不回显

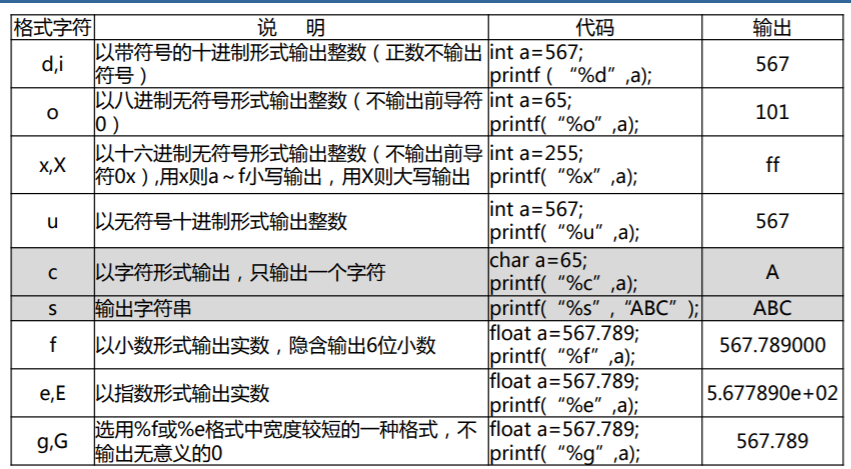
getche()回显于屏幕

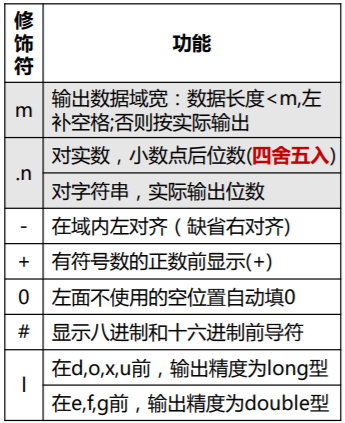
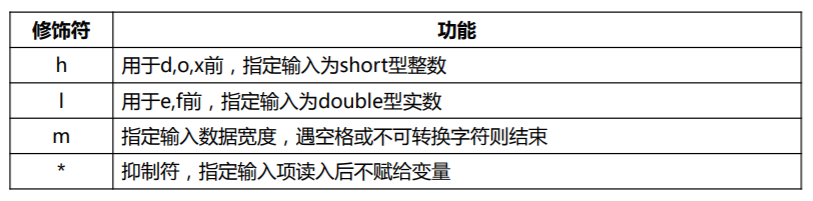
缓冲型（按回车回车才有反应）：混合型scanf（）

字符型getchar（）gets（）

Output 混合型：printf（）

字符型：putchar（）puts（）

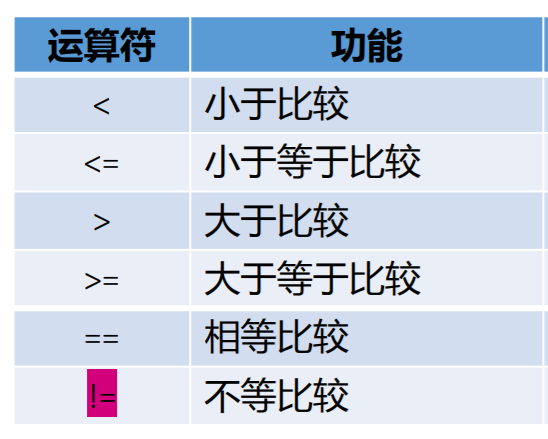




**注意：1/2得到的是0**

**1.0/2得到的是0.5**

**第三节课**



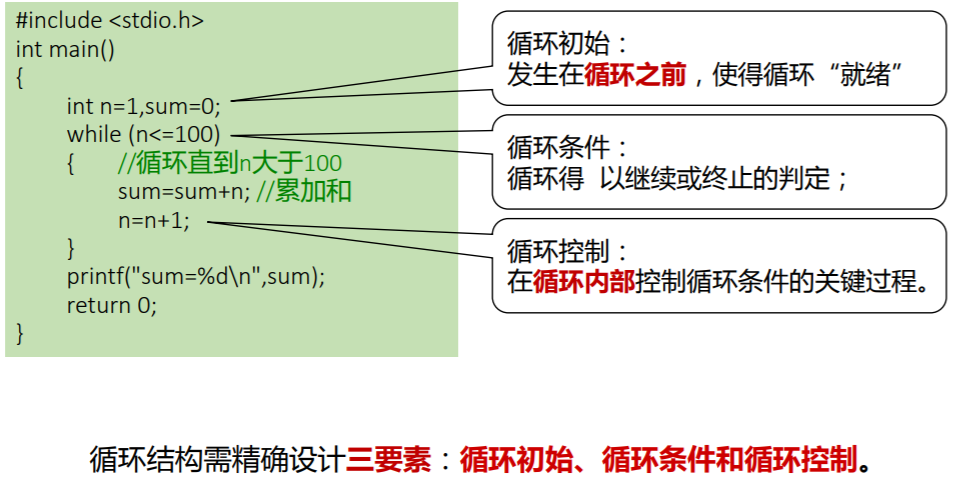
双等号判断是否相等

浮点型不用等号做相等和不等运算

Expr！=0和expr等价都是判断真假的意思

expr1 ? expr2 : expr3

“case常量表达式”只是起语句标号作用，并不做条件判断。应该在执行一个case分支后,可以用一个break语句终止执行switch语句。**循环结构·**



**第四节课**

数组：

int A[10]； A包含十个整型数据

长度是整型、符号或常量表达式

EOF=-1

排序题（起泡法）

字符串多留一位

循环找空格时，循环条件限定一个“i!=’\0’”

Puts输出自动回车？

