



知识点总结



- C语言中，数据类型转换有**隐式转换（自动类型转换）**和**显式转换（强制类型转换）**两种方式。

- **隐式转换**：发生在不同类型的数据进行混合运算或赋值操作时，由编译器自动完成。

- 遵循以下规则：

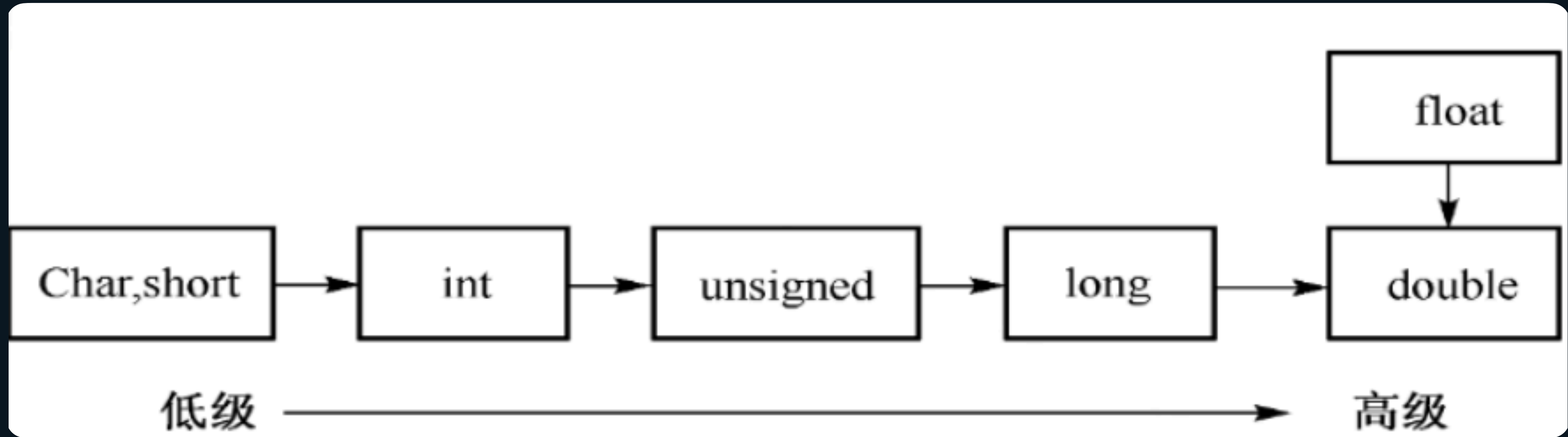
- 1) 若参与运算的两个数据的类型不同，则先转换成同一类型，再进行运算。

- 2) 转换遵循从低精度类型到高精度类型的原则，以保证精度不降低。

如：int类型和double类型运算时，先把int类型转换成double类型后，再进行运算。

- 3) 在赋值操作中，等号两边的数据类型不同时，等号右边的类型将转换为左边的类型。如果右边数据的范围超过左边类型的表示范围，超出的部分会被截断。

如：int类型和double类型运算时，先把int类型转换成double类型后，再进行运算。





- **显式转换**：将一种数据类型强制转换成另一种数据类型。
- 语法格式：**(数据类型) 值/变量/式子**

例子：

```
int main() {  
    int var_1 = (int)4.9; // (数据类型)值  
    double var_2 = (double)var_1; // (数据类型)变量  
    int var_3 = 5;  
    double result = (double)(var_1 * var_3); // (数据类型)(式子)  
}
```

- 注：式子也要用一对括号括起来



- 无论是**隐式转换**还是**显式转换**，都只是对变量进行**临时**的类型转换，并不会真的改变变量原本定义的数据类型和实际值。

例子：

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

```
int main() {  
    double var = 9.8;  
    // 打印结果: var as int = 9  
    printf("var as int = %d\n", (int)var);  
    // 打印结果: var as double = 9.800000  
    printf("var as double = %lf\n", var);  
}
```



题目1、有以下程序：

```
#include<stdio.h>
int main()
{
    int a;
    a = (int)1.99 * 3;
    printf("%d\n", a);
}
```

程序的运行结果是（ A ）。

A. 3

B. 5

C. 6

D. 结果不确定



题目2、设整型变量 $a = 2$ ，则执行下列语句后，浮点型变量 b 的值不为0.5的是 (**B**)

- A. $b = 1.0 / a$
- B. $b = (\text{float})(1 / a)$
- C. $b = 1 / (\text{float})a$
- D. $b = 1 / (a * 1.0)$



题目3、若 “`int n; float f = 13.8;`”,则执行 “`n = (int)f % 3`” 后, n 的值是 (A)

A. 1

B. 4

C. 4.333333

D. 4.6



题目4、 $(\text{int})((\text{double})9 / 2) - 9 \% 2$ 的值是 (B)

A. 0

B. 3

C. 4

D. 5



题目5、若在程序中变量均已定义成int类型，且已赋大于1的值，则下列选项中，能正确表示代数式 $1 / abc$ 的结果的是 (A)

A. $1.0 / a / b / c$

B. $1 / (a * b * c)$

C. $1.0 / a * b * c$

D. $1 / a / b / (\text{double})c$