第四章学习心得

自2001班 郭锴

.成员访问运算符

点运算符和箭头运算符都可访问成员，ptr->mem等价于(\*ptr).men。解引用优先级低于运算符。

条件运算符（？：）cond?expr1:expr2cond为真对expr1求值并返回，否则对expr2求值

可以嵌套a=(grade>90)?"high pass":(grade<60)?"fail":"pass"

输出表达式使用条件运算符，记得加最外层的括号cout<<((grade<60)?"fail":"pass");

位运算符

\*\*建议仅将位运算符用于处理无符号类型。\*\*因为位运算符如何处理运算对象的符号位依赖于机器，而且左移操作可能改变符号位位置。移位运算符：左移<<右移>>求反运算符：~

位与&，位或|，异或^

sizeof运算符sizeof返回一个表达式或者一个类型名字所占的字节数。所得的值是一个size\_t类型的常量表达式。\*\*sizeof并不会实际计算其运算对象的值，\*\*所以再sizeof运算对象中解引用一个无效指针依然安全。如sizeof \*p。sizeof(引用类型)得到被引用对象所占空间大小。sizeof(指针)得到指针本身所占空间大小。sizeof(解引用指针)得到指针指向的对象所占空间大小，指针不需要有效。sizeof(数组)得到数组所占空间大小，注意不会将数组转换为指针处理。数组大小/单个元素大小=数组元素个数，constexpr size\_t size=sizeof(arr)/sizeof(\*arr)sizeof(vector/string对象)只返回该类型固定部分的大小，不会计算对象中的元素占了多少空间。

类型转换

1.算术转换

运算符的运算对象会转换成最宽的类型。整型提升：小整数类型转换为大整数类型。（例如bool, signed char, unsigned char, char, short,unsigned short会被提升为int）

2.其它隐式类型转换

数组自动转换成指向数组首元素的指针，数组被用作decltype关键字参数，或者取址符，sizeof，typeid等运算符的运算对象时，上述转换不会发生。同时用引用初始化数组时也不会发生。算术类型或指针类型自动转换为布尔类型。

指向非常量的指针转换为指向相应常量类型的指针，引用也是。

3.显式转换

强制类型转换：cast\_name <目标类型> （表达式）static\_cast:只要不包含底层const就可以

void\* p=&d; //任何非常量对象的地址都可以存入void \*double \*dp=static\_cast<double\*>(p)； //将void\* 类型转换为初始指针类型\*\*const\_cast:\*\*去const性质reinterpret\_cast旧式类型转换：（目标转换类型）表达式

\*\*尽量避免强制类型转换。