**四，表达式**

1，基础

小整数类型（如bool,char,short等）通常会被提升成较大的整数类型，主要是int。

C++的表达式要不然是右值，要不然就是左值。左值可以位于赋值语句的左侧，右值则不能。

当一个对象被用作右值的时候，用的是对象的值（内容）；当对象被用作左值的时候，用的是对象的身份（在内存中的位置）。

一个重要的原则是在需要右值的地方可以用左值来代替，但是不能把右值当成左值使用。

使用关键字decltype的时候，左值和右值也有所不同。如果表达式的求值结果是左值，decltype作用于该表达式（不是变量）得到一个引用类型。

2，算术运算符

3，逻辑和关系运算符

4，赋值运算符

赋值运算符的左侧运算对象必须是一个可修改的左值。

5，递增和递减运算符

除非必须，否则不用递增递减运算符的后置版本。后置版本需要将原始值存储下来以便于返回这个未修改的内容。造成性能浪费。

6，成员访问运算符

ptr->mem等价于(\*ptr).mem。

7，条件运算符

8，位运算符

位运算符提供检查和设置二进制位的功能。

位运算符：~位求反，<<左移，>>右移，&位与，^位异或，|位或。

9，sizeof运算符

sizeof运算符返回一条表达式或一个类型名字所占的字节数。

求数组中元素的个数constexpr size\_t sz = sizeof(ia) / sizeof(\*ia);

10，逗号运算符

11，类型转换

虽然有时不得不使用强制类型转换，但这种方法本质上是非常危险的。

命名的强制类型转换包含了static\_cast,dynamic\_cast,const\_cast和reinterpret\_cast。

任何具有明确定义的类型转换，只要不包含底层const，都可以使用static\_cast。

cosnt\_cast只能改变运算对象的底层const。

reinterpret\_cast通常为运算对象的位模式提供较低层次上的重新解释。

旧式的强制类型转换有两种形式：type(expr)和(type)expr

根据所涉及的类型不同，旧式的强制类型分别具有与const\_cast,static\_cast或reinterpret\_cast相似的行为。

12，运算符优先级表