第十一章：

1. 通过对运算符的重载可以实现在你运算符的对象为类的对象时，进行对类的运算。例如，重载+使得两个矢量相加，重载+使得两个时间相加等运算。重载运算符可以方便对类的使用，但是重载运算符有相对的限制——要至少一个定义的类型和要符合原来的句法规则。同时不能修改运算顺序。
2. 在重载运算符中，遇到了进行乘法计算时，编译器严格按照你输入的句法顺序而不能实现乘法交换律的问题，对于这个问题，需要调用私有成员本身，这在成员函数中是非法的，于是引入友元函数可以通过调用对象本身来对\*运算符进行重载。利用友元函数也可以达到重载<<运算符达到输出类的目的。
3. 在接受唯一一个参数的构造函数可以被c++用作类的转换函数，但在该构造函数前加上关键字 explicit 可以禁止该功能。

第十二章

1. 在类的构造函数中可以使用new关键字来达到节约内存的作用。在类中使用static关键字可以创造静态储存成员，该成员为同类的所有对象共享的一个参数，该对象不能再类声明中进行初始化，可以在类声明以外的单独语句进行初始化，初始化指出类型和作用域运算符，没有使用关键字static。在使用构造函数中使用new时，需要在析构函数中使用delete，同时二者需要互相兼容，即new对应delete，new【】对应delete【】，因此有多个构造函数时，需要使用相同的new格式。同时需要定义复制构造函数和赋值运算符以达到深复制的目的。
2. 在创造类时，c++将自动提供以下成员函数，默认构造函数、默认析构函数、复制构造函数、赋值运算符和地址运算符。在使用c++默认的默认析构函数和复制析构函数时，因为赋值的对象赋值在一个新对象，在复制指针时，会复制两个指向相同地址的指针，因此在析构函数中使用delete时会破坏该指针指向的值，使用在使用时需要重新定义该函数以达到深复制的目的。