1. 在使用函数时常常用到形参，形参肯定要用到拷贝，当拷贝的类型为一个很搭的数据时将会非常消耗性能，此时，我们可以使用引用来避免拷贝。

举例：当传递一个很长的字符串时，为了减少性能消耗可以使用引用

int func(string &a)

{

return a.size();

}

我们可以发现，在这个函数中并没有对string进行操作，从安全和性能方面考虑，我们可以把函数参数改成const string &a，这对函数的功能完全没有影响。所以，可以得出这样一个结论：

如果函数无需改变引用形参的值，最好将其声明为常量引用。

1. 尽量使用常量引用

把函数不会改变的形参定义成引用是一种比较常见的错误，这么做会导致函数的调用者误以为可以一次去修改实参的值。此外，使用引用而非常量引用也会极大地限制函数所以能接受的实参类型，比如，我们不能把const对象、字面量或者需要类型转换的对象传递给普通的引用形参。