

函数重载是在一定作用域内，多个相同名称但不同参数列表的函数重载  
编译时由编译器根据实际调用时给的实参情况来判定本次实际用哪个函数， 这个过程叫重载决策  
重载函数本质上就是多个独立函数， 重载机制在编译时发生，运行时不参与  
函数重载的意义就是避免我们胡乱起名，方便编写类库覆盖所有可能操作，是一种语法  
表面上，运算符重载是对 C++ 源生运算符的意义，在某个 class 中做重定义本质上，运算符被映射到执行相应的成员函数，所以运算符重载其实是重定义对象的运算符所对应的函数  
运算符重载是一种语法特性，C++ 全面支持，Java 不支持，python 有限度的支持  
没有运算符重载照样写代码，所有操作全部通过显式调用相应成员函数来完成即可运算符重载是一种语法糖，可以让代码“看起来更简洁，更优雅”，将复杂实现隐藏在类的内部  
运算符重载机制加大了类库作者的工作量，减少了调用类库写功能的人的书写量  
C++ 支持运算符重载机制有其理念和历史原因，是对写法简洁和效率优秀的综合考量  
赋值运算符 = 重载和引用结合，可有效提升代码效率  
赋值运算符 = 重载时要注意有指针成员时会涉及浅拷贝和深拷贝  
运算符重载一定程度上体现了 C++ 的多态性，因为同样的运算符在不同的 class 中表现是不同的