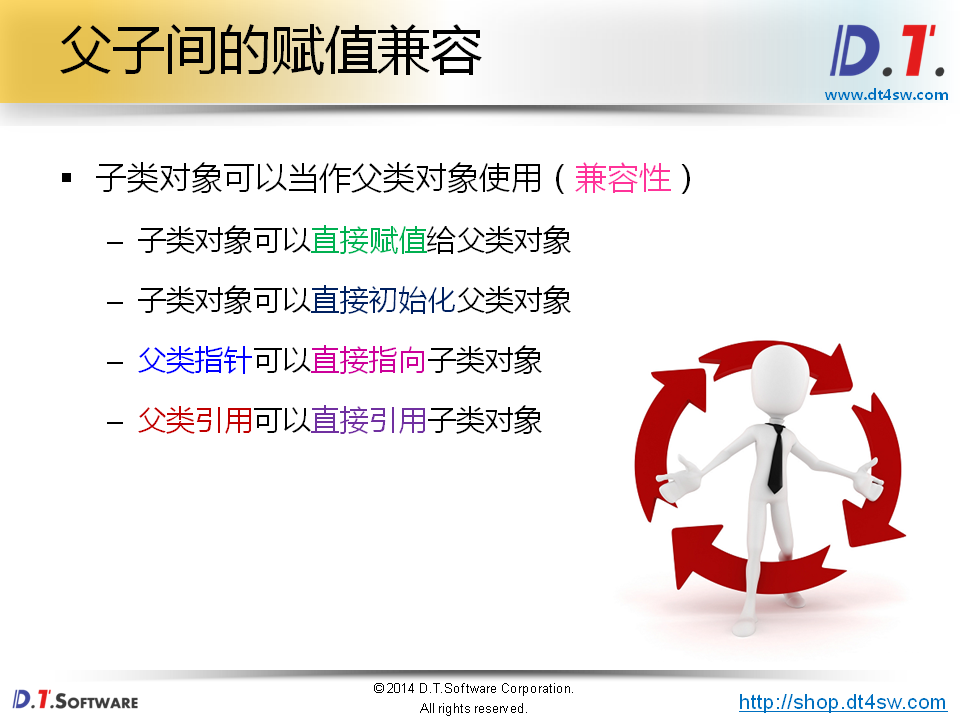
理解基类与派生类之间的类型转换是理解C++面向对象编程的关键。

# 派生类及其对象向基类的类型转换

因为在派生类对象中含有与其基类对应的组成部分，所以我们能把派生类的对象当成基类对象来使用，可能将基类的指针或者引用绑定到派生类对象中的基类部分上，在此期间执行赋值兼容原则。



在派生类对象中含有与其基类对应的组成部分，这一事实是继承的关键所在。

# 为什么基类指针（引用）可以绑定到派生类上？

当使用基类的指针（引用）时，我们并不清楚该指针（引用）所绑定对象的真实类型。因为动态类型直到运行时才可知。

## 动态类型与静态类型：

静态类型：变量声明时的类型或表达式生成的类型。

动态类型：变量或表达式表示的内存中的对象的类型。

# 不存在从基类向派生类的隐式类型转换

之所以存在派生类向基类的类型转换是因为每个派生类对象都包含一个基类部分，而基类的引用或指针可以绑定到该基类部分上。如果将派生类指针（引用）绑定到基类上，在运行时有可能将指针（引用）指向动态类型中基类没有的部分。

# 使用派生类对象给基类对象赋值

当我们用一个派生类对象为一个基类对象初始化或赋值时，只有该派生类对象中的基类部分会被拷贝，移动或赋值，它的派生类部分会被忽略掉。