|  |  |
| --- | --- |
| 1. 模板实参推断 2. 类型转换与模板类型转换 3. 使用相同模板参数类型的函数形参 4. 正常类型转换应用于普通函数实参 | Const转换  数组或函数指针转换  Template <typename T> T fobj(T,T)  Template <typename T> T fref(const T&,const T&)  Const string s2();  Fobj(s1,s2);  Fref(s1,s2);  flexibleCompare(lng,1024)  print(cout,42);  ofstream f(“output”);  print(f,10) |
| 1. 模板实参推断 2. 从函数实参来确定模板实参的过程被称为模板实参推断 3. 编译器生成的函数版本与给定的函数调用最为匹配 4. 类型转换与模板类型转换   T是按值传递的，从值推断类型   1. 编译器不是对实参进行类型转换，而是生成一个新的模板实例 2. 顶层const在形参还是实参都会被忽略。（注意：是当函数形参是T，不是T&时） 3. 类型转换中，有两种能应用于函数模板 4. Const转换：一个非const对象的引用传递给一个const的引用(或指针)形参 5. 数组或函数指针转换：若函数形参不是引用，可以对数组或函数类型的实参应用正常的指针转换。(也没说：若是引用，不能对数组到指针的转换)   其它的转换都不能应用于函数模板   1. 使用相同模板参数类型的函数形参 2. 用两个T为函数形参，则实参必须是相同类型 3. 正常类型转换应用于普通函数实参 4. 函数模板可以有用普通类型定义的参数。他们正常转换为对应形参的类型   注意：  先执行实参的类型是否相同，再执行推断与转换T | |