

.值传递：有一个形参向函数所属的栈拷贝数据的过程，如果值传递的对象是类对象或是大的结构体对象，将耗费一定的时间和空间。  
  
2.指针传递：同样有一个形参向函数所属的栈拷贝数据的过程，但拷贝的数据是一个固定为4字节的地址。  
  
3.引用传递：同样有上述的数据拷贝过程，但其是针对地址的，相当于为该数据所在的地址起了一个别名。

一.静态数据成员  
1.静态数据成员的定义。  
  
静态数据成员实际上是类域中的全局变量。所以，静态数据成员的定义(初始化)不应该被放在头文件中,应放到对应的cpp文件。  
其定义方式与全局变量相同  
不要在头文件中定义(初始化)静态数据成员。在大多数的情况下，这样做会引起重复定义这样的错误。即使加上#ifndef #define #endif或者#pragma once也不行  
2.静态数据成员被 类 的所有对象所共享，包括该类派生类的对象。即派生类对象与基类对象共享基类的静态数据成员。  
3.静态数据成员(静态函数)可以成为成员函数的可选参数，而普通数据成员（函数）则不可以。  
4.静态数据成员的类型可以是所属类的类型，而普通数据成员则不可以。普通数据成员的只能声明为所属类类型的 指针或引用。  
  
二．静态成员函数：  
  
1.静态成员函数的地址可用普通函数指针储存，而普通成员函数地址需要用 类成员函数指针来储存。  
2.静态成员函数不可以调用类的非静态成员。因为静态成员函数不含this指针。  
3.静态成员函数不可以同时声明为 virtual、const、volatile函数。