多重继承总结：

1. 在C++中所有的继承都是实现继承，因为在一个基类、接口和实现中的任何内容都将成为派生类的一部分，只继承一个类的某些部分（比如只继承接口）是不可能的。接口继承只是在一个派生类接口中加入了成员函数的声明，因此在C++中模拟接口继承常见的技术是从一个仅包含声明（没有数据和函数体）的接口类派生一个类。除了析构函数以外，这些声明都是纯虚函数。
2. 测试函数可以接受任何一种类型的参数，这些参数类型与正确的识别标志和返回类型一起提供了一个成员函数的定义；这些参数并不必要派生自一个共同的基类。
3. 多重继承的一个共同用途包括使用混入类，混入类的存在是为了通过继承来增加其他类的功能，它不能刻意地由它本身进行实例化。
4. 真正的菱形继承是Left和Right子对象在一个完整的Bottom对象内部共享着一个Top对象。这是通过使Top成为Left和Right的一个虚基类实现的。
5. Left和Right子对象各有一个指向共享的Top子对象的指针，并且对Left和Right成员函数中那个子对象的所有引用都要通过这些指针来完成。
6. 平常避免使用多重继承。但有些虚函数的问题可以由多重继承解决。