1. **简述什么是函数重载，调用重载函数时要注意哪些问题。**

答：函数名相同但参数列表却不相同，参数列表的不同有三种含义：

参数个数不同，或参数类型不同，或者参数个数和类型都不同。

函数重载与默认参数同时出现，容易引起函数调用的二义性。

1. **分别解释C++中的两种参数传递机制。**

答：C++中有两种参数传递机制：值传递和引用传递。

值传递：作为实参的表达式的值被复制到由对应形参名所标识的一个对象中，作为形参的初始值，函数对形参的访问、修改都是在这个标识对象上操作的，与实参无关，即数据的传递是单向的；

引用传递：使用引用作为函数形参时，调用函数的实参要用变量名，实参传递给形参，相当于在被调用函数中使用了实参的别名，于是在被调函数中对形参的操作实质是对实参的直接操作，即数据的传递是双向的。

1. **从定义格式、作用和调用时机三方面辨析构造函数与析构函数的异同点。**

答：

构造函数：构造函数是一种特殊的成员函数，对象的创建和初始化工作可以由他来完成，它在创建对象时被自动调用。其特点有：函数名与类名相同，可以重载，不能指定返回类型，不能被显式调用。

析构函数：析构函数也是一种特殊的成员函数，它的作用是在对象消失时执行清理任务，它的格式为：<类名>::~<类名>() { 函数体 }，它在释放时被自动调用，析构函数不能有参数，所以不能重载，一个类中只能定义一个析构函数，不能指定返回值。函数名为波浪符加类名。可以被显式调用但不推荐调用。

1. **简述赋值兼容规则。**

答：实质在公有继承的情况下，一个派生类的对象可以作为基类的对象来使用。具体来说,就是以下3种情况：

派生类的对象可以赋给基类的对象。

派生类的对象可以初始化基类的引用。

派生类的对象的地址可以赋给指向基类的指针。

1. **简述在继承方式下创建派生类对象时，构造函数调用顺序，以及删除派生类对象时派生类析构函数的调用顺序。**

答：首先基类构造函数被调用，子对象所在类的构造函数次之，最后执行派生类构造函数。删除对象时析构函数的调用顺序正好相反。

1. **从现存类中派生出新类时，可以对派生类做哪些变化？**
2. 增加新的成员变量
3. 增加新的成员函数
4. 重新定义已有的成员函数
5. 改变现有成员的属性
6. **分别解释C++中类的三种访问控制。**

答：C++中对于类的成员具有Public、Protected和Private三种访问控制。

Public：公有成员定义了类的外部接口，在类外均可访问Public的成员。

Protected和Private分别是保护的和私有的成员，私有成员是被隐藏的数据，只有该类的成员函数或类的友元函数才可以访问它们。保护成员具有公有成员和私有成员的双重性质，可以被类的或其派生类的成员函数或友元函数访问。

1. **什么叫做多态性？C++如何实现多态的？**

答：多态性就是同一符号或名字在不同情况下具有不同解释的现象，即是指同一个函数的多种形态。C++支持两种多态性，编译时的多态性和运行时的多态性。

多态性有两种表现形式：

1. 不同的对象在收到相同的消息时产生不同的动作，这主要通过虚函数来实现；

2. 同一对象收到相同的消息却产生不同的函数调用，这主要通过函数重载来实现。

1. **动态联编需要满足三个条件:**
2. 类之间满足类型兼容规则；
3. 要声明虚函数；
4. 要由成员函数来调用或者是通过指针、引用来访问虚函数。