1. 一些开发环境自动生成的代码里在构造函数前出现explicit关键字，它的作用是什么？

在C++中，当一个类的构造函数只有一个参数时，可以通过直接赋值的方式来对对象进行初始化。如“Number number = 5;”这样的形式。Number是一个类，5是一个int型的值。能够这样做的前提是Number这个类中有一个参数为整型的构造函数。这种现象称之为构造函数的隐式调用。

在构造函数的声明时加上explicit关键字，可以关闭构造函数的隐式调用。

1. 讲一讲析构函数的作用

析构函数是用于在撤销对象时完成对**对象的清理工作**，比如在创建对象时，如果在构造函数中动态申请了内存，那么在对象释放时，应该在析构函数中对动态申请的内存进行释放，避免造成内存泄露，如果在这个时候还不释放就没有机会释放内存了，会造成内存泄露，总之需要在释放对象之前完成的工作都可以放在析构函数中完成，析构函数不需要用户显示调用，它会在释放对象前由系统自动调用。

1. 析构函数能不能手动调用呢？调用后能不能结束对象的生命周期

析构函数可以手动调用，但是一般不推荐这样做。因为析构函数中主要是对象中的资源在对象生命周期结束时进行回收。而手动调用会导致对象的部分资源在生命周期结束前就已经回收的现象。可能导致段错误。

而析构函数只是在对象生命周期结束后调用，单纯地调用析构函数不能结束对象的生命周期。

1. 构造函数初始化时通过列表初始化和在函数体内初始化有什么区别？用那种好？

这和对象在创建时的步骤有关，对象创建时

第一步、会先申请一块内存空间，

第二步、会对这块空间进行初始化，

第三步、最后会调用构造函数。

列表初始化会在第二步执行，而构造函数在第三步执行。从效率上来说列表初始化更优。所以尽可能使用列表初始化。

1. this这个指针是在哪里声明的

所有成员函数和构造析构函数都有一个隐藏参数，会把该对象的地址以参数的形式传入函数。这个地址在使用的时候就是this指针了。

Car(int wheel ,string color)

1. new/delete和malloc/free的区别

delete会调用对象的析构函数,而free只会释放内存，new调用构造函数。malloc与free是C++/C语言的标准库函数，

new/delete是C++的运算符。它们都可用于申请动态内存和释放内存。对于非内部数据类型的对象而言，光用malloc/free无法满足动态对象的要求。对象在创建的同时要自动执行构造函数，对象在消亡之前要自动执行析构函数。由于malloc/free是库函数而不是运算符，不在编译器控制权限之内，不能够把执行构造函数和析构函数的任务强加于malloc/free。因此C++语言需要一个能完成动态内存分配和初始化工作的运算符new，以及一个能完成清理与释放内存工作的运算符delete。注意new/delete不是库函数。

1. delete与 delete [ ] 区别

delete只会调用一次析构函数，而delete[]会调用每一个成员的析构函数。在More Effective C++中有更为详细的解释：“当delete操作符用于数组时，它为每个数组元素调用析构函数，然后调用operator delete来释放内存。”delete与New配套，delete []与new []配套

1. 类和对象的存储区域是怎样的

类是存放在代码段当中，唯一。

对象视创建的方式而定，使用new运算符创建的存放在堆中，直接使用构造函数创建的存放在栈中

1. 类的静态成员和非静态成员有何区别

静态成员为所有类的实例对象共享，静态成员有静态成员变量和静态成员函数，静态成员变量使用前必须初始化，静态成员变量可以被静态成员函数和非静态成员函数访问，而静态成员函数只能访问静态成员变量，因为静态成员函数属于类，其没有this指针。非静态成员每个对象都有一个。

1. 为什么静态成员函数不能访问非静态的类成员

静态成员函数和类绑定，只要有类就可以调用，而非静态的成员和对象绑定，有对象才能访问。有类的时候未必就存在对象