**C++面向对象程序设计复习提纲**

1. **C++一个更好的C**

函数重载 ：规则，目的在于提高程序可读性

内联函数 ：规则，目的在于提高函数调用速度

引用 ：语法，参数传递和返回值

参数默认值 ：规则

1. **类和对象**

类类型，是由用户定义的自定义类型。类是创建对象的样板，是对具有相同属性的对象的描述，封装(包含)着对象的数据描述和操作的定义。对象，是类的实例，是一组属性和行为的集合，是数据与操作的集合，封装是c++中的重要特性。

对象成员的可访问性规则 ：

本类成员函数内可访问本类的所有成员、基类的保护和公有成员、友元类的成员；

本类成员函数外，只可访问对象的公有成员（包括本类和从基类继承的）、类的公有静态成员、友元类的所有成员；

聚合，是类之间的包含关系，一个类中可以包含另外一个类的对象作为数据成员（也称为子对象）；

构造函数的初始化列表用于指定基类或成员的构造；初始化列表没有指定时，代表调用基类或成员的缺省（无参）构造函数。

对象可通过对象名访问，也可通过对象指针或引用访问；同一类型的多个对象可用数组表示；

在每一个成员函数（静态成员函数除外）中都包括一个特殊的指针，指向自身对象，称为this；它的值是当前被调用的成员函数所在对象的起始地址。由此决定了静态成员函数不能访问本类中的非静态成员；

常成员函数用const修饰，不可修改类成员，包括间接修改，const也可用来修饰参数或常量

特殊类机制 ：静态数据成员和静态函数成员，静态成员语义效果相当于全局变量；友元函数和友元类

3. **对象的构造和析构**

对象生成时自动调用类的构造函数，进行初始化；销毁时，自动调用类的析构函数进行扫尾处理；**全局对象**在程序开始执行时生成，在程序执行结束时销毁；**局部对象**在程序执行对象所在的复合语句（函数体）时在栈上生成，在所在复合语句执行完毕离开时或抛出异常离开时销毁；**动态分配对象**在new时生成，在delete时销毁或负责管理的智能指针对象销毁时销毁；作为成员的**子对象**随所在**宿主对象**一起生成和销毁

1. **参数传递和对象复制、赋值**

在函数间，C++对象（或普通变量）可通过指针传递间接访问，也可通过引用访问，通过传值访问时，会调用类的复制构造函数复制一个对象的副本操作；在函数返回时，局部对象会销毁，所以，不可返回局部对象的引用或地址；对象的复制通过类的复制构造函数实现，编译器提供的是浅层复制，简单对象可直接使用编译器提供的浅层复制构造函数即可，涉及动态分配的复杂对象需重载复制构造函数，实现深层复制；赋值时道理类同，简单对象可直接使用编译器提供的浅层赋值函数即可实现赋值，涉及动态分配的复杂对象需重载赋值运算符，实现深层赋值；C++支持临时对象的移动构造和移动赋值，极大提高效率。std::move (x)可将x视作临时对象。

5. **运算符重载**

C++运算符重载主要有成员运算符重载和友元运算符重载2种方式；运算符重载时优先级不变、结合性不变、操作数个数不变；大多数运算符均可重载，极少量运算符不可重载；运算符重载不可改变标准类型上的意义，也不可引入新运算符；重载后++、--时为与前++、--区分人为引入一个无用整形参数；对象重载[]运算符后可像数组一样通过[]访问，重载()运算符后可像函数一样使用，称为函数对象或仿函数；

一个参数的构造函数可起到参数类型到对象类型的转换作用，对象类型到其它类型的转换借助于重载转换运算符实现。

6. **函数模板和类模板**

目的在于提高代码的通用性；

函数模板：实际上是建立一个通用函数，其函数类型和参数类型在函数定义时不具体指定，用一个虚拟的类型来代表， 推迟到函数调用时才确定（显式指定或由编译器推定），这个通用函数就是函数模板。C++支持模板的特化和部分特化。

类模板是类的抽象，模板类是类模板的实例。

1. **继承和多态性**

继承，创建新类的一种方法，是由一般类到特殊类的过程。

派生类对象构造时先执行基类构造，再执行派生类构造。基类的构造在初始化列表中指定，未指定时调用基类缺省构造函数；析构时现执行派生类析构再执行基类析构；

赋值兼容原则 ： 基类对象的引用或指针可接收（代表）派生类对象

派生类继承了基类功能，基类函数不同名时可直接使用函数名调用，同名时存在同名覆盖原则，需通过基类名：：成员函数名调用；

C++支持多继承，多继承中多个基类的成员同名问题可通过基类名：：成员名解决，多继承中菱形继承导致的多份成员的冗余和不一致问题需借助于虚继承实现；

多态性，指一对多的状态，比如函数重载，一个函数名对应不同的实现和功能，又比如动态联编，一般类中的行为或方法被特殊类继承后，可有不同的实现或操作，并在运行中进行联编。编译时的多态性称静态多态性，运行时多态性称动态多态性。动态多态性通过虚函数体现，多个类需有一个共同基类，多态性通过使用基类对象的引用或指针调用虚函数实现；C++用纯虚函数表示还没有实现的虚函数，含纯虚函数的类称为抽象类。

1. 异常

C++支持在必要时抛出异常，在外部按声明匹配顺序捕捉异常；异常可分类，标准库异常有一个共同基类 ：exception，它有一个虚函数what返回异常信息；

1. 典型容器模板vector,string,stack,queue的常用接口的意义和应用、类（模板）的实现方法（技术）。
2. STL容器的迭代器访问方法，迭代器可看作泛型指针，所指对象类型为容器中元素类型；数组名可看作数组的迭代器；STL容器对象通过迭代器构造；STL典型算法count、count\_if、find find\_if、sort的用法；函数对象的定义和使用方法。

**典型程序 ： 历次作业**