### C++

* 1. 每个对象拥有自己的数据成员，但类的成员函数只有一份，哪个对象调用成员函数，成员函数就作用在哪个对象上。
  2. This指针的使用，哪个对象调用该函数，this指针就指向哪个对象。
     1. This指针只能在成员函数中使用，全局函数、静态函数都不能使用this。
  3. 如果一个函数参数设定了缺省值时，其右边的参数都要有缺省值。
  4. 重载函数与默认参数共同使用时，要注意出现二义性问题
  5. 无参的构造函数与全是默认参数的重载构造函数不能同时出现。
  6. 构造函数类型
     1. 默认构造函数
     2. 重载构造函数
     3. 带参数列表的构造函数：成员变量（参数）
  7. 拷贝构造函数也是构造函数，写法：point（const point &obj）;(是默认拷贝构造函数，point p2=p1、point p3(p1)、对象作为函数参数（传值）、返回临时对象（类外函数类型是类）)
     1. 默认拷贝构造函数是浅拷贝，堆对象不能调用浅拷贝构造函数，不然析构函数会对同一资源释放两次。
     2. 在浅拷贝构造函数里加一行this->count++；(静态成员函数)就成了深拷贝了。？
        1. 浅拷贝：this->x = obj.x;this->x = obj.x;可忽略
  8. 对于析构函数，对象销毁自动调用，对象在堆必须delete掉
  9. 深拷贝构造函数在有指针成员数据的时候一定要自己写个。
     1. 有指针对象的时候可以浅拷贝其他对象的数据，因为该指针对象已有自己一套指针成员数据
     2. 指针对象在类外函数delete;指针成员必须自己写个析构函数delete
  10. 定义的类对象只能访问公有成员，不能对象.私有成员，必须通过公有函数对象才能间接访问私有成员；类外函数只能通过类对象才能访问类。
  11. 类外函数属于所有人，类内函数属于对象，静态函数属于类（static void test()，对象可以调用，但会有隐患）。
      1. 静态成员函数不能直接访问非静态变量或函数，非静态成员函数可以访问静态成员。
      2. 静态数据成员用法
         1. Static int count；统计类创建对象个数(声明)
            1. 一定要在全局中初始化：int Point::count=0;然后在构造函数中count++;
  12. 常对象：const 类名 对象/类名 const 对象。常熟组：类型 const 数组名[n].
      1. 常成员函数定义：类::函数名 () const。
         1. 声明 void show() const;
      2. 其他 常变量、常引用、常指针都是const 类型 （）
  13. 静态成员函数不能是常函数?
  14. 运算符重载实现了类的多态性
      1. . \* ? : sizeof不能重载
      2. Point operator+(const Point &obj);声明
         1. 定义 Point Point::operator +(const Point &obj){}
      3. 先加后用：Point operator++();声明
      4. 先用后加：Point operator++(int );
  15. 输入输出只通过友元函数对运算符重载
      1. 类外函数（不加类名限定符）、是类的朋友（可以访问类的私有成员）、打破了类的封装性
      2. friend ostream& operatorn <<(ostream &cout,const Point &obj)声明
         1. ostream& operatorn <<(ostream &cout,const Point &obj){}
      3. friend istream& operatorn >>(istream &cin, Point &obj)
      4. 友元类：把所有函数都声明为友元
  16. 继承
      1. 可以继承父类空间大小
      2. 类外的访问方式：对象只能访问公有继承的公有成员
      3. 公有成员：类外可见，类内可见，派生类可见
      4. 保护成员：类外不可见，类内可见，派生类可见
      5. 私有成员：类外不可见，类内可见，派生类不可见
      6. 私有继承的类内函数不可以访问父类私有，可以访问本类私有，他有自己一个和父类继承的两个私有。