1. C++引入了什么
   1. 面向对象的思想
   2. 泛型编程
2. 面向对象的三大特征
   1. 封装，继承，多态
   2. 封装：把数据和方法抽象出来，放在一起
   3. 继承：代码复用，扩展类的新功能
   4. 多态：可以实现接口，可以不动用前人写的代码来扩展新功能
      1. 多态成立条件：有继承，重写父类的虚函数，父类指针指向子类对象
3. 名字控制
   1. namesapace
4. 结构体的加强
   1. 结构体里可以写方法，定义变量时不用写struct
5. 更加严格的类型转换
6. C++的三目运算符
   1. C++的三目运算符返回的是左值，C的三目运算符返回的是右值
7. const
   1. C语言的const修饰的变量都有内存
   2. C++语言的const修饰的变量有时没有内存，有时有内存
      1. 什么时候没有内存：发生常量折叠时
   3. 尽量用const替代define
      1. 因为define没有类型，不重视作用域
8. 引用
   1. 引用是给空间取别名
   2. 引用的内部是用常指针实现
   3. 引用传递的是地址
   4. 引用是编译器帮我们做了取地址和解引用
   5. 指针引用：给指针变量取了别名
   6. 常引用：字面量不能赋给引用，因为没有空间。const的修饰的可以
9. 内联函数
   1. 内联函数可以有宏函数的效率，有函数的优势，我们只能向编译器申请，同意是否成为内联函数是编译器说的算
10. 函数的默认参数和占位参数
    1. 函数定义时，给参数初始化。
       1. 默认参数注意点：如果有默认参数，那么该参数后面的参数必须都是默认参数
       2. 以传入的值优先，如果没传入值，就用默认的值
    2. 占位参数在后置加加时使用
11. 函数重载
    1. 同一个作用域：函数名相同的函数之间的关系叫重载
    2. 条件：参数个数不同，参数的类型不同，参数的顺序不同
    3. 原理：编译器给函数名取了别名
12. C++调用C语言的函数：extern “C”
13. 类
    1. 装载数据和方法的结构体
    2. 结构体和类的区别：结构体的默认权限是公有，类的默认权限是私有
    3. 类里成员的权限：公有，私有，保护
    4. 成员属性尽量私有化
    5. 什么函数编译器会给我们加:默认构造，默认析构，默认的拷贝构造，默认的赋值函数
    6. 构造函数和析构函数的调用顺序
       1. 在对象创建时，调用构造函数，在对象销毁前调用析构函数
       2. 继承中：先调用父类的成员对象的构造函数，然后调用父类构造函数，再调用子类成员对象的构造函数，然后调用自己的构造函数，析构函数调用顺序相反。
    7. 拷贝构造函数调用的时机
       1. 用一个对象去初始化另一个对象
       2. 做函数参数时
       3. 函数局部对象以值的方式返回出去，有的调用，有的不调用
       4. 深浅拷贝：
          1. 浅拷贝：使用编译器提供的默认拷贝构造函数，造成同一块空间被释放两次。
          2. 深拷贝：解决浅拷贝问题，1.自己提供拷贝构造函数，2.申请空间，拷贝数据
       5. 赋值函数：
          1. 如果调用编译器提供的赋值函数，那么内存泄漏，同一块空间被释放两次，解决的方法：自己提供赋值函数，1.释放原空间，2.申请空间，拷贝数据
    8. 初始化列表
       1. 当类里有成员对象时，成员对象没有提供默认的构造函数，那么要用初始化列表来显示的调用哪个构造函数
    9. 动态创建对象
       1. new和delete
       2. new对象时：分配空间，调用构造函数
       3. delete:先调用析构函数，释放空间
       4. new时有[],那么delete时也要[]
    10. 静态成员
        1. 静态成员变量
           1. 属于类，不属于对象，但是是所有的对象共享
           2. 静态成员变量在类内声明，类外定义，生命周期为整个程序，作用域在类内
        2. 静态成员函数
           1. 管理封装静态成员变量，不能访问普通成员变量（没有this指针）
    11. 单列模式
        1. 一个类只能实例化一个对象
           1. 把构造函数私有化（拷贝构造）
           2. 在类内定义一个私有的静态成员指针
           3. 在类外new个对象给他
           4. 在类内提供获取该指针的方法
    12. 类里的成员变量和成员函数是分开存储
    13. this指针
        1. 每个对象都有一个编译器给的this指针
        2. this指针的作用：1.当函数参数的名字和成员变量名字。返回对象本身
    14. const修饰的成员函数叫常函数
        1. 常函数内不能改变成员变量，但是mutable修饰的成员变量除外
    15. 常对象
        1. const修饰的对象叫常对象
        2. 常对象只能调用常函数，不能修改成员变量，但是mutable修饰的除外
    16. 友元
        1. 友元是访问类的私有成员
        2. 友元函数，友元类，友元的成员函数
    17. 运算符重载
        1. 让代码的可读性提高
        2. 运算符重载的本质：函数调用
        3. 函数名统一:operator
        4. 重载前置加加好，为什么，后置加加需要定义临时变量
    18. 多继承
        1. 不建议使用多继承，同名成员产生二义性问题
        2. 菱形继承：解决菱形继承的方法，用虚继承，这时基类叫虚基类
    19. 多态
        1. 多态是干什么的，可以不用动前人的代码，来添加新功能，开闭原则
        2. 多态发生的条件：继承，重写父类的虚函数，父类指针指向子类对象
        3. 静态联编和动态联编：静态联编在编译阶段就确定了函数调用地址，动态联编是在运行的时候确定函数的调用地址
        4. 使函数成为虚函数，告诉编译器进行动态联编（晚绑定）
        5. 抽象类：类里有纯虚函数的类叫抽象类，抽象类不能实例化对象
        6. 如果子类继承了抽象类，那么必须重写所有的纯虚函数，不然也变成抽象类
        7. 虚析构函数
           1. 父类指针指向子类对象，让父类指针调用子类的析构函数
           2. 纯虚析构函数：需要再类外实现，
           3. 纯虚析构函数会让类变成抽象类