1. C++概述
   1. C++两大编程思想
      1. 面向对象 泛型编程
   2. C++98标准
2. C++书写hello world
   1. 包含头文件 #include <iostream> 标准输入输出头文件
   2. using namespace std; 使用标准命名空间
   3. cout << “helloworld” << endl; end line;
   4. 面向对象三大特性 ：封装、继承、多态
3. 双冒号作用域运算符
   1. ::如果前面没有任何作用域，代表了使用全局作用域
4. namespace 命名空间
   1. 用途：解决命名冲突
   2. 命名空间下可以存放 变量、函数、结构体、类…
   3. 命名空间 只能存放在全局作用域下
   4. 命名空间可以嵌套命名空间
   5. 命名空间是开放的，可以随时添加新成员
   6. 命名空间是可以匿名的
   7. 命名空间可以起别名
5. using声明和using编译指令
   1. using声明
      1. using kingGlory::sunwukongId;
      2. 如果出现就近原则，需要避免两个原则同时出现
   2. using编译指令
      1. using namespace kingGlory;
      2. 如果出现就近原则，优先使用就近原则
      3. 如果有多个房间打开，并且存在同名的变量，使用时候需要加作用域区分
6. C++对C语言增强
   1. 全局变量检测增强 int a ; int a = 10;
   2. 函数检测增强
      1. 声明函数返回值
      2. 形参类型 检测
      3. 返回值检测增强
      4. 调用时候 参数传参个数检测增强
   3. 类型检测增强 malloc 返回void \* C++下必须强转
   4. struct 增强
      1. C++可以放入函数
      2. 使用时C++可以省去关键字 struct
   5. bool类型增强
      1. C++才有bool类型
      2. sizeof 1
      3. true --- 1 代表真 false --- 0假
   6. 三目运算符
      1. C语言下 返回的值
      2. C++语言下返回的是 变量
   7. const增强
      1. C语言下
         1. 全局const 语法可以通过，运行报错
         2. 局部const 语法通过 ，运行成功
      2. C++语言下
         1. 全局const 语法可以通过，运行报错
         2. 局部const 语法通过，运行修改不成功，因为变量放入到符号表中
   8. const在C语言下 默认是外部链接属性
   9. const在C++语言下，默认是内部链接属性
   10. const分配内存情况
       1. 对于变量 取地址
       2. 在变量前加关键字 extern
       3. 使用变量 来初始化 const修饰的变量
       4. 对于自定义数据类型，也会分配内存
   11. 尽量利用const来代替#define
       1. 宏常量 没有类型
       2. 不重视作用域
7. 引用
   1. 用途 给内存起别名
   2. 语法：type &别名 = 原名
   3. 引用必须要初始化
   4. 一旦初始化后，就不可以修改指向了
   5. 建立对数组的引用
8. 参数的传递方式
   1. 值传递
      1. 不可以修改到本体
   2. 地址传递
      1. 可以修改到本体
   3. 引用传递
      1. 可以修改到本体
9. 注意事项
   1. 不要返回局部变量的引用
   2. 如果函数的返回值是引用，那么这个函数调用可以作为左值进行运算
10. 引用的本质
    1. 就是一个指针常量
    2. 所以引用必须要初始化
    3. 初始化后 不可以修改指向
11. 指针的引用
    1. 通过引用技术 可以简化指针
12. 常量的引用
    1. 使用场景：修饰函数中的形参，防止误操作