1 C++引入面对对象的思想及泛型编程

2 面向对象的三大特征：封装，继承，多态

3 C++引入名字的控制

4 输出流，输入流，换行及刷新缓冲区

5 作用域，：：

6 名字的控制

6.1 namespace创建命名空间

6.2 可以扩展

6.3 只能全局

6.4 可以嵌套

6.5 可以取别名

6.6 里面可以有方法

6.7 无命名空间

6.8 分文件编写，要注意命名空间

7 Using声明和编译指令

7.1 Using A::a，需要注意重定义

7.2 usingnamespace A;就不需要注意重定义

8 结构体加强

8.1 C语言的结构体不能定义方法

8.2 C++的结构体可以定义方法，而且定义变量时不需要加上struct关键字

9 C++必须显示的类型转换

10 C语言的三目运算符返回的是右值，C++的三目运算符返回的是左值

11 Const

11.1 C语言const具有外部链接属性

11.2 C语言const修饰的变量都有内存

11.3 C++中const修饰的全局变量具有内部链接属性，要改变用extern

11.4 C++connst修饰的变量有时有内存，有时没有内存

11.5 常量折叠问题

11.6 常量折叠

11.6.1 编译器只能折叠基础数据类型，而且在编译阶段不知道变量的值

11.6.2 volatile是告诉编译器不要优化

12 尽量用const替代define

12.1 Define没有类型，const有类型

12.2 Define不重视作用域

13 引用

13.1 引用的本质：是内存空间的别名

13.2 引用的作用：使语法更加简洁

13.3 引用实际传递的是什么：地址

13.4 引用的注意：

13.4.1 引用的指向一旦初始化，就不能改变

13.4.2 引用必须初始化

13.4.3 引用必须保证引用合法的内存空间

13.5 指针的引用和C语言中的什么相似，二级指针。

13.6 引用实际是编译器帮我们做了哪两步：\*a(解引用)，&a(取地址)