**目标**

复习C++对象的概念

掌握构造函数的基本用法

掌握析构函数的基本用法

理解数据共享的方式

掌握静态成员的使用方法

了解共享数据的保护形式

掌握友元的使用

掌握Const关键字的详细用法

继承与派生讲解

1、复习C++对象的概念

C++ 对象，根据类来进行创建，类能够定义一些相同事物的共有的属性和行为。员工->Employee 类

2、掌握构造函数的基本用法

在为一个对象分配好内存后，通过构造函数给内存中的属性名进行赋值。

//雇佣了一个新的员工

Employee emp(“张三”,26,”程序员”);

3、掌握析构函数的基本用法

析构函数：当不用一个对象时，如何清理对象使用过的所有内存资源。

//解雇一个人的时候，把分配给他的所有资源，全部收回

~Employee();

4、理解数据共享的方式

作用域：程序中通过 { } 符号，表示的一个 范围。

class Employee{

private:

string name; //姓名

double salary; //工资

public:

//计算工资

double getSalary(int year){

double sal = 0.0;

if(year>0 && year <1){sal = 3000;} //不满一年，工资 3000

if(year>=1 && year <2){sal = 5000;} //不满两年，工资 5000

if(year>=2 && year <5){sal = 7000;} //

retrun sal;

}

}

{} 作用域已经能够控制部分属性的访问范围。C++ 中还补充了一些新的修饰符号、单词。

static

const

friend

private

protected

public

5、掌握静态成员的使用方法

C 程序如何使用内存：代码区、栈区、堆区、常量区(普通常量区、静态常量区)

static : 在一个类加载完成，还没有运行之前。给修饰的 量分配内存。

对一个类来说，可以通过类名 或 对象名进行访问。不管用那种方式访问，实际上访问的都是同一块内存。任何人都可以修改，但是修改会影响到其他所有使用 该属性的代码。

Static 关键字 修饰属性和方法：

(1)修饰属性

static int age; //表示属性的值存入常量区，所有对象共享同一个值

(2)修饰方法

static void xxx();//表示该方法只能通过类进行访问，不能通过对象访问

//static 修饰的方法，可以访问 static 修饰的属性，并修改他们的值

让一个类的所有对象，共享同一块内存。

6、了解共享数据的保护形式

private： 私有修饰符，只能在类内部使用，无法在外部访问，子类中也不能访问。

Protected: 半公开的修饰符，只有在 父子类之间 该修饰 公开的。

Public : 对所有的类，全部公开

7、掌握友元的使用

friend 朋友，C++ 中叫友元。

两个完全无关的类，可以通过修饰符 friend ，在特定的情况下，让一个类可以访问另外一个类中所有的属性(private protected public)

8、掌握Const关键字的详细用法

修饰常量： const 修饰的常量，是在分配完内存，并调用的时候才真正存入数据。一个对象，只能使用自己的 const 值，不能多个对象用一个 const 值。

Const 使用方式有四种：

(1)修饰属性

int const PI=3.1415926; //永远不变

(2)修饰函数的输入参数

void print(const int a); //在 print{} 中 a 的值不能被修改

(3)修饰函数返回值

const int Area(); //表示调用该函数后，返回值是不能被改变的

(4)修改整个函数本身

int A::show() const; //表示整个函数中，只能调用不能对其他属性做任何修改,只能看不能改

9、继承与派生讲解

类：是一些具有相同属性和行为的 对象的 模版。

Apple 🡪 iPhone (触摸屏幕、上网、安装 App、拍照)

Google ->

三星 ->

小米 ->

我们如果开发的话，不能单独的为 Apple 设计手机，写一个类

class IPhone{

//属性

private:

屏幕尺寸

处理器

内存

摄像头

//行为

打电话

上网

看电影

照相

};

class Note2{

//属性

private:

屏幕尺寸

处理器

内存

摄像头

//行为

打电话

上网

看电影

照相

};

class Mi{

//属性

private:

屏幕尺寸

处理器

内存

摄像头

//行为

打电话

上网

看电影

照相

};

问题：如果有多种类型事物，编写代码的时候，每个类中都写同样的代码。代码没有办法重复利用。

解决：在类的基础上，创建一个新的 “代码复用”的机制---继承。

不用重复发明 “轮子”

交通工具(启动、运行、停止)

|

--------------------------------------------------------------

| | | |

汽车 火车 船(漂浮、舵) 飞机

轿车、卡车 高铁 普快 货车 游船 邮船 直升机

C++ 中继承分为两种：单一继承、多重继承

形状(周长、面积) 输入设备 输出设备

| | |

----------- ------------------

| | |

长方形 ……. 圆形 输入输出设备