

C++程序设计第六节课官方笔记

目录

- 一、 课件下载及重播方法
- 二、 本章/教材结构图
- 三、 本章知识点及考频总结
- 四、 配套练习题
- 五、 其余课程安排

一、 课件下载及重播方法

二、 教材节构图



三、 本章知识点及考频总结

(一) 选择题 (共 12 道)

1. 静态局部变量具有局部作用域，但却具有全局生存期。也就是说，静态局部变量在程序的整个运行期间都存在，它占据的空间一直到程序结束时才释放，但仅在定义它的块中有效，在块外并不能访问它。

静态变量均存储在全局数据区，静态局部变量只执行一次初始化。如果程序未显式给出初始值，则相当于初始化为 0；如果显式给出初始值，则在该静态变量所在块第一次执行时完成初始化。

2. 类的静态成员有两种：静态成员变量和静态成员函数。在类体内定义类的成员时，在前面添加 **static 关键字** 后，该成员即成为**静态成员**。

类的静态成员被类的所有对象共享，不论有多少对象存在，静态成员都只有一份保存在公用内存中。对于静态成员变量，各对象看到的值是一样的。

定义类静态成员变量时，在类定义中声明静态成员变量，然后必须在类体外定义静态成员变量的初值。这个初值不能在类体内赋值。

给静态成员变量赋初值的格式如下：

```
类型 类名::静态成员变量=初值；
```

注意，在类体外为静态成员变量赋初值时，前面不能加 **static 关键字**，以免和一般的静态变量相混淆。在类体外定义成员函数时，前面也不能加 **static 关键字**。

访问静态成员时，成员前面既可以用类名作前缀，也可以使用对象名或对象指针作前缀。这与访问类成员时仅能使用对象名或对象指针作前缀是不同的。

访问类静态成员的一般格式如下：

```
类名::静态成员名
```

或是

```
对象名.静态成员名
```

或是

```
对象指针->静态成员名
```

类的静态成员函数没有 **this** 指针，不能在静态成员函数内访问非静态的成员，即通常情况下，类的静态成员函数只能处理类的静态成员变量。静态成员函数内也不能调用非静态成员函数。

3. 对于普通成员变量，每个对象有各自的一份，而**静态成员变量只有一份，被同类所有对象共享**。普通成员函数一定是作用在某个对象上的，而静态成员函数并不具体作用在某个对象上。

4. 对于静态变量，如果没有进行初始化，系统会自动初始化为 0。局部变量如果没有进行初始化，则其值是不确定的。

使用 **new** 运算符创建的变量具有动态生存期。从声明处开始，直到用 **delete** 运算符释放存储空间或程序结束时，变量生存期结束。

5. 类的常量成员变量必须进行初始化，而且只能通过构造函数的成员初始化列表的方式进行。使用 **const** 修饰的函数称为常量函数。定义类的对象时如果在前面添加 **const** 关键字，则该对象称为常量对象。

定义常量对象或常量成员变量的一般格式如下：

```
const 数据类型 常量名=表达式；
```

定义常量函数的格式如下：

```
类型说明符 函数名（参数表）const；
```

在对象被创建以后，其常量成员变量的值就不允许被修改，只可以读取其值。对于常量对象，只能调用常量函数。总之，常量成员变量的值不能修改，常量对象中的各个属性值均不能修改。

说明常量对象后，不能通过常量对象调用普通成员函数。

6. 一个类的成员变量如果是另一个类的对象，则该成员变量称为“**成员对象**”。这两个类为包含关系。**包含成员对象的类叫作封闭类。**

在定义封闭类的构造函数时，需要添加初始化列表，指明要调用成员对象的哪个构造函数。在封闭类构造函数中添加初始化列表的格式如下：

```
封闭类名::构造函数名(参数表): 成员变量 1(参数表), 成员变量 2(参数表), ...  
{...}
```

初始化列表中的成员变量既可以是成员对象，也可以是基本数据类型的成员变量。对于成员对象，初始化列表的“参数表”中列出的是成员对象构造函数的参数（它指明了该成员对象如何初始化）。对于基本数据类型的成员变量，“参数表”中的就是初始值。“参数表”中的参数可以是任何有定义的表达式，该表达式中可以包括变量甚至是函数调用等，只要表达式中的标识符都是有定义的即可。

7. **友元的概念破坏了类的封装性和信息隐藏，但有助于数据共享，能够提高程序执行的效率。**

8. 友元使用关键字 **friend** 标识。在类定义中，当 **friend** 出现在函数说明语句的前面时，表示该函数为类的友元函数。一个函数可以同时说明为多个类的友元函数，一个类中也可以有多个友元函数。当 **friend** 出现在类名之前时，表示该类为类的友元类。

9. 不能把其他类的私有成员函数声明为友元函数。

友元函数不是类的成员函数，但允许访问类中的所有成员。在函数体中访问对象成员时，必须使用“对象名.对象成员名”的方式。

友元函数不受类中的访问权限关键字限制，可以把它放在类的公有、私有、保护部分，结果是一样的。

10. 如果将一个类 B 说明为另一个类 A 的友元类，则类 B 中的所有函数都是类 A 的友元函数，在类 B 的所有成员函数中都可以访问类 A 中的所有成员。在类定义中声明友元类的格式如下：

```
friend class 类名;
```

友元类的关系是单向的。若说明类 B 是类 A 的友元类，不等于类 A 也是类 B 的友元类。**友元类的关系不能传递**，即若类 B 是类 A 的友元类，而类 C 是类 B 的友元类，不等于类 C 是类 A 的友元类。

11. C++语言规定，当调用一个成员函数时，系统自动向它传递一个隐含的参数。该参数是一个指向调用该函数的对象的指针，称为 **this 指针**，从而使成员函数知道对哪个对象进行操作。

C++规定，在非静态成员函数内部可以直接使用 **this** 关键字，**this** 就代表指向该函数所作用的对象的指针。

在一般情况下，在不引起歧义时，可以省略“**this->**”，系统采用默认设置。

12. 静态成员是类具有的属性，不是对象的特征，**this** 表示的是隐藏的对象指针，所以静态成员函数没有 **this** 指针。

(二) 主观题 (共 0 道)

四、配套练习题

1、下列关于静态数据成员的特性叙述中，错误的是 ()

A:说明静态数据成员时，使用关键字 **static** 进行修饰

B:静态数据成员要在类外进行初始化

C:引用静态数据成员时，要在静态数据成员名前加<类名>和作用域运算符

D:静态数据成员是所有对象的共享成员

2、下列关于对静态数据成员的描述中，正确的是（ ）

A:静态数据成员不能用 public 控制符修饰

B:静态数据成员可以直接用类名或者对象名来调用

C:静态数据成员不可以被类的对象调用

D:静态数据成员不能用 private 控制符修饰

3、友元函数的主要作用是（ ）

A:提高程序的效率

B:加强类的封装性

C:实现数据的隐蔽性

D:增加成员函数的种类

[参考答案] DBA

五、其余课程安排