

学习内容总结

当需要使用 `NULL` 时候，请养成直接使用 `nullptr` 的习惯。

`constexpr` 表示常数表达式，`constexpr` 的函数可以使用递归。

C++11 支持 `auto` 和 `decltype` 关键字进行类型推导。

C++11 使用 `std::initializer_list`，允许构造函数或其他函数像参数一样使用初始化列表。

C++11 使用 `using` 引入了下面这种形式的写法，并且同时支持对传统 `typedef` 相同的功效。

C++11 加入了新的表示方法，允许任意个数、任意类别的模板参数，同时也不需要再在定义时将参数的个数固定。

定义了变长的模板参数，可以用 `sizeof`、递归模板函数、初始化列表（`Lambda` 表达式和 `std::initializer_list`）展开解包。

C++11 支持委托构造，即构造函数可以在同一个类中一个构造函数调用另一个构造函数。

C++11 利用关键字 `using` 引入了继承构造函数的概念。

C++11 引入了 `override` 和 `final` 关键字来防止意外重载虚函数的发生。

`override` 关键字将显式的告知编译器进行重载

`final` 防止类被继续继承以及终止虚函数继续重载引入的。

c++11支持显式禁用默认函数

```
1 Magic() = default; // 显式声明使用编译器生成的构造
2 Magic& operator=(const Magic&) = delete; // 显式声明拒绝编译器生成构造
```

C++11 引入了枚举类（enumeration class），并使用 `enum class` 的语法进行声明，实现了类型安全。

学习心得：

了解了更多的c++11的新特性，特别是 `Lambda` 表达式的运用十分灵活。

待解决问题：

无。

明日计划：

继续学习c++，ceph集群配置。