

个人资料



zjufirefly



访问：6408次

积分：463

等级：BLOG > 2

排名：千里之外

原创：42篇 转载：4篇

译文：0篇 评论：1条

文章搜索

文章分类

C++ (14)

其他 (5)

链接 (13)

linux (13)

shell (7)

操作 (11)

文章存档

2014年12月 (8)

2014年11月 (7)

2014年10月 (8)

2014年09月 (6)

2014年08月 (4)

展开

阅读排行

git操作命令 (327)

blktrace (287)

mysql常用命令 (280)

python (270)

cgroups (262)

linux下如何配置openvpn (260)

2015年4月微软MVP申请

《京东技术解密》有奖试读，礼品大放送

"我的2014"年度征文活动火爆开启

CSDN 2014博客之星

C++语法杂项1

分类：C++

2014-05-06 22:53

89人阅读

评论(0)

收藏

编辑

删除

1.拷贝构造函数必须为引用或有常量引用，否则编译无法通过。
因为是普通的传值，需要先构造临时对象，会再次调用拷贝构造函数，造成递归。所以拷贝构造函数的入参不能为普通传值。

2.typename

```
template<class T>
class MyClass{
    typename T::SubType *ptr;
};
```

typename指出SubType是Class T中定义的一个型别，ptr是一个指向T::SubType型别的指针。
没有typename，SubType会被当成Class T类中的一个static成员。

定义内部型别

```
class Q {
    typedef int SubType;
}
```

```
class Q {
    class SubType;
};
```

3.非型别模板
例如：bitset<20> flags20;

4.缺省模板参数

```
template <class T, class container = vector<T> >
class MyClass;
```

5.成员模板

```
template<class T>
class MyClass {
    T value;
    template<class X>
    void assign(const MyClass<X>& x);
}
```

Template Constructor是成员模板的一种特殊形式。Template constructor通常用于“再复制对象时实现隐式型别转换”。
注意：template constructor并不隐藏implicit copy constructor。

vim-ctags-taglist-netrw	(249)
性能测试命令字段解释	(213)
正则表达式	(211)
SystemTap	(195)

评论排行	
为什么需要auto_ptr_ref	(1)
valgrind内存检查	(0)
cgroups	(0)
linux下如何配置openvpn	(0)
git操作命令	(0)
thrift	(0)
Apache开源软件	(0)
vim-ctags-taglist-netrw	(0)
vim常用配置	(0)
shell按行读取字符串，并	(0)

推荐文章	
* 挣扎与彷徨——我的2014	
* 校招回忆录——小米篇	
* Android UI-自定义日历控件	
* 30岁程序员回顾人生、展望未来	
* 2014年终总结，我决定要实现的三个目标	
* Android 启动问题——黑屏 死机解决方法	

最新评论	
为什么需要auto_ptr_ref	
zjufirefly: explicit auto_ptr_ref(_Tp1* __p): _M_ptr(__p) {}...	

6.异常抛出

void f() throw();// 不抛出任何异常
void f() throw(bad_alloc, bad_cast);// 抛出bad_alloc或bad_cast异常
void f(); // 可能抛出任何异常

7.namespace

using declaration:using std::cout;
using directive:using namespace std;
定义名字空间:
namespace josuttis {
class File;
}

8.如果某个函数的一个或多个参数型别，定义与函数所处的名字空间中，那么可以不为改函数指定名字空间，这个规则称为**Koenig lookup**

9.reinterpret_cast

如何使用？

10.常数静态成员初始化

```
class MyClass{  
static const int num = 100;  
};  
const int MyClass::num;// 需要定义一个空间，VS2005不进行定义也可以正常运行。  
TODO：确认标准如何规定
```

11.非常量静态成员的初始化

```
class MyClass{  
public:  
static int num ;  
};  
int MyClass::num = 100;// 需要定义一个空间
```

12.异常分类

语言本身支持的异常：
bad_alloc,bad_cast,bad_typei,bad_exception
C++程序库所发生的异常：
logic_error,及其之类invalid_argument,length_error,out_of_range,domain_error
程序作用域之外发生的异常：
runtime_error及其之类range_error,overflow_error,underflow_error
程序库所抛出的异常：
range_error,out_of_range,invalid_argument,bad_alloc及客户代码中抛出的异常。

13.const char* p, char* const p

const char* p p指向的内如不能改变。
char* const p 指针p不能改变。p不能再指向其他数据。

14.只有当对象完全构造成功，才有可能于将来调用其构造函数。

ClassB (Class A& val1, ClassA& val2):ptr1(new ClassA(val1),ptr2(new ClassA(val2)){存在资源泄露可能，解决方法是使用智能指针。

15.数值极限

头文件： <limits>

范例：numeric_limits<int>::min()

上一篇

list中如何查找两个元素间的某个元素

下一篇

常用头文件

主题推荐

c++

智能指针

namespace

exception

对象

猜你在找

C++编程规范之13确保资源为对象所拥有使用显示的

2013级C++第6周春项目对象指针const项目1-体验常

C++智能指针-TR1的shared_ptr和weak_ptr使用介绍

C++基础积累lusing namespace potter 自定义命名

c++中关于智能指针std::shared_ptr的用法

C++ 工程实践1慎用匿名 namespace

C++智能指针-TR1的shared_ptr和weak_ptr使用介绍

c++智能指针

C++基础之TR1库的智能指针

C++智能指针简单剖析

1 C语言

2 博客搬家工具

3 c++

4 c9

5 商品条码

6 生肖银币

7 女式服装加盟

8 切骨机

9 紫薇苗圃

10 新疆翻译

11 ewin棋牌官网

12 红宝石蛋糕加盟

13 欧洲买房移民

14 北京会计网首页

15 绿之韵官方网站

16 加盟养生多少钱

17 水果捞加盟

18 1000炮街机捕

19 净化工程公司

20 移民中介机构

查看评论

暂无评论

发表评论

用户名：

zjufirefly

评论内容：

提交

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-600-2320

北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved

blog.csdn.net/zjufirefly/article/details/25163473

3/3