



博客园 首页 新随笔 联系 订阅 管理

## C++ 流插入"<<"和流提取">>"运算符的重载

微信公众号：「小林coding」  
用简洁的方式，分享编程小知识。

### 01 流插入<<运算符的重载

C++ 在输出内容时，最常用的方式：

```
std::cout << 1 <<"hello";
```

问题：

- 那这条语句为什么能成立呢？
- cout 是什么？ "<<" 运算符能用在 cout 上呢？

原因：

- 实际上，cout 是在 iostream 头文件中定义的 ostream 类的对象。
- "<<" 能够用在 cout 上是因为，在 ostream 类对 "<<" 进行了重载。

对于 std::cout << 1 <<"hello"; 这条语句，有可能按以下方式重载成 ostream 类的成员函数：

```
ostream & ostream::operator<<(int n)
{
    .... // 输出n整型的代码
    return *this;
}

ostream & ostream::operator<<(const char * s)
{
    .... // 输出s字符串的代码
    return *this;
}
```

- std::cout << 1; 语句，等价于 cout.operator<<(1);
- std::cout << "hello"; 语句，等价于 cout.operator<<("hello");
- std::cout << 1 <<"hello"; 语句，等价于 ( cout.operator<<(1) ).operator<<("hello");

### 02 流插入<<运算符重载的例子

假定我们想把某个对象里的内容进行打印输出，那么我们可以重载 ostream 类的流插入 << 运算符。

下面以 CStudent 类作为例子：

```
class CStudent // 学生类
{
public:
    // 构造函数
    CStudent(int id = 0, int age = 0, string name = ""):m_id(id), m_age(age), m_name(name) {}

    // 将该函数声明成友元函数
    // 目的是使得函数可以访问CStudent类的私有成员变量
    friend ostream & operator<<(ostream & o, const CStudent & s);

private:
    int m_age; // 年龄
    int m_id; // ID号
    string m_name; // 名字
};

// 重载ostream对象的流插入<<运算符函数
// 目的是使得能打印输出CStudent对象的信息
ostream & operator<<(ostream & o, const CStudent & s)
{
    o << s.m_id << ", " << s.m_age << ", " << s.m_name;
    return o;
}

int main()
{
    CStudent stu(1, 20, "小林coding");
    std::cout << stu; // 输出stu对象的全部信息

    return 0;
}
```

输出结果：

```
1,20,小林coding
```

需要注意的是 ostream & operator<<(ostream & o, const CStudent & s) 函数是全局的，所以函数的第一个参数必须要传入 ostream 的对象，并且 CStudent 类需要将该函数声明成友元函数，使得函数可以访问 CStudent 类的私有成员变量。

### 03 流提取>>运算符重载的例子

还是以 CStudent 类作为例子，假设想通过键盘的输入的内容，来初始化对象，则我们可以重载 istream 类的流提取 >> 运算符。

```
class CStudent // 学生类
{
public:
    // 构造函数
    CStudent(int id = 0, int age = 0, string name = ""):m_id(id), m_age(age), m_name(name) {}

    // 将该函数声明成友元函数
    // 目的是使得函数可以访问CStudent类的私有成员变量
    friend ostream & operator<<(ostream & o, const CStudent & s);

    // 将该函数声明成友元函数
    // 目的是使得函数可以给CStudent类的私有成员变量进行赋值
    friend istream & operator>>(istream & is, CStudent & s);

private:
    int m_age; // 年龄
    int m_id; // ID号
    string m_name; // 名字
};

// 重载ostream对象的流插入<<运算符函数
// 目的是使得能打印输出CStudent对象的信息
ostream & operator<<(ostream & o, const CStudent & s)
{
    o << s.m_id << ", " << s.m_age << ", " << s.m_name;
    return o;
}

// 重载istream对象的流提取>>运算符函数
// 目的是使得初始化CStudent对象的内容
istream & operator>>(istream & is, CStudent & stu)
{
    string inputStr;
    is >> inputStr;

    int pos = inputStr.find(", ", 0); // 查找首次出现逗号的位置
    string tmpStr = inputStr.substr(0, pos); // 截取从0到pos位置的字符串
    stu.id = atoi(tmpStr.c_str()); // atoi可以将char*类型的内容转成int类型

    int pos2 = inputStr.find(", ", pos + 1); // 查找第二次出现逗号的位置
    tmpStr = inputStr.substr(pos + 1, pos2 - pos - 1); // 取出age的值
    stu.age = atoi(tmpStr.c_str()); // atoi可以将char*类型的内容转成int类型

    tmpStr = inputStr.substr(pos2 + 1, inputStr.length() - pos2 - 1); // 取出name的值
    stu.name = tmpStr;

    return is;
}

int main()
{
    CStudent stu;

    // 将输入的信息，初始化stu对象
    cin >> stu;

    // 输出stu对象的信息
    cout >> stu;

    return 0;
}
```

输入内容和输出内容：

```
// 输入内容：
1,20,小林coding

// 输出内容：
1,20,小林coding
```

### 04 小结

要想流插入 << 运算符和流提取 >> 运算符能针对自定义的对象，那么我们就需要重载针对该对象的 ostream 类的 << 运算符和 istream 的 >> 运算符，并且只能重载成全局的函数，然后在 CStudent 类里需要把上面的两个重载函数声明成友元函数，使得两个重载的函数可以访问和赋值 CStudent 类里的私有成员函数。



2019 2020

小林coding

关注微信 一起成长

关注公众号：「小林coding」，回复「我要学习」即可免费获得「服务器 Linux C/C++」成长路程（书籍资料 + 思维导图）

标签：C++ 基础知识, C/C++

好文要顶

关注我

收藏该文

小林coding  
关注 - 6  
粉丝 - 458  
+加关注

0 推荐 0 反对

« 上一篇：C++ 手把手教你实现可变长的数组  
» 下一篇：C++ 自增、自减运算符的重载和性能分析

posted @ 2019-12-02 12:27 小林coding 阅读(2786) 评论(1) 编辑 收藏 举报

最新评论 刷新页面 返回顶部

登录后才能查看或发表评论，立即 登录 或者 逛逛 博客园首页

编辑推荐：  
· 戏说领域驱动设计（五）——子域  
· 什么是向上管理  
· 从 MVC 到 DDD 的架构演进  
· ASP.NET Core 6 框架揭秘实例演示[02]：基于路由、MVC 和 gRPC 的应用开发  
· 理解 ASP.NET Core - 基于 JwtBearer 的身份认证

百度智能云 开发者上云优惠专场 云服务器0元/月

最新资讯：  
· 以色列将建立独立自主的量子计算能力：开发本国首台量子计算机  
· 月薪8千离职赔偿100万？深度起底“宁王”黄业协议  
· 发改委文引引导外卖平台下调商家佣金，美团股价今日暴跌16%  
· 员工称因“泄露工资”被开除，盒马鲜生最新回应，薪资保密是否合法？  
· 大众汽车收购华为自动驾驶业务？已有华为员工收到“内部转岗约谈”  
» 更多新闻...

## 公告



公众号回复「网络」，免费获取自制 300 页图解网络 PDF  
公众号回复「系统」，免费获取自制 300 页图解系统 PDF  
公众号回复「pdf」，免费获取热门计算机书籍

昵称：小林coding  
年龄：2年6个月  
粉丝：458  
关注：6  
+加关注

2022年2月						
日	一	二	三	四	五	六
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	1	2	3	4	5
6	7	8	9	10	11	12

## 搜索

找我看

谷歌搜索

## 我的标签

- 图解计网(20)
- C/C++(20)
- 图解操作系统(19)
- C++基础知识(15)
- Linux(11)
- 小鸟初学Shell编程(9)
- 图解MySQL(6)
- 图解Redis(4)
- Shell脚本(3)
- 学习心得(2)
- 更多

## 随笔档案

- 2022年2月(1)
- 2022年1月(4)
- 2021年12月(2)
- 2021年9月(4)
- 2021年8月(3)
- 2021年7月(4)
- 2021年6月(1)
- 2021年5月(2)
- 2021年4月(2)
- 2021年2月(3)
- 2021年1月(2)
- 2020年12月(1)
- 2020年11月(1)
- 2020年10月(4)
- 2020年9月(4)
- 更多

## 阅读排行榜

- C++ 线程安全的单例模式总结(26883)
- 30张图解：TCP 重传、滑动窗口、流量控制、拥塞控制(24362)
- C++ 深入浅出工厂模式（进阶篇）(24114)
- Linux shell 获得字符串所在行数及位置(17137)
- 原来 8 张图，就可以搞懂「零拷贝」了(16474)

## 评论排行榜

- 硬核！30 张图解 HTTP 常见的面试题(24)
- 完蛋，公司被一条 update 语句干趴了！(21)
- 30张图解：TCP 重传、滑动窗口、流量控制、拥塞控制(21)
- 为了掌握 Redis 数据结构，我画了 40 张图（完整版）(19)
- 原来 8 张图，就可以搞懂「零拷贝」了(17)

## 推荐排行榜

- 为了掌握 Redis 数据结构，我画了 40 张图（完整版）(78)
- 硬核！30 张图解 HTTP 常见的面试题(66)
- 看书的一点小建议！(45)
- 原来 8 张图，就可以搞懂「零拷贝」了(41)
- 拔掉网线后，原本的 TCP 连接还存在吗？(39)

## 最新评论

- Re:拔掉网线后，原本的 TCP 连接还存在吗？  
博主很优秀哦，给你点赞  
——窥视未来
- Re:拔掉网线后，原本的 TCP 连接还存在吗？  
学习  
——民工也Coding
- Re:拔掉网线后，原本的 TCP 连接还存在吗？  
这个好  
——人间春风意
- Re:拔掉网线后，原本的 TCP 连接还存在吗？  
厉害  
——leslie\_xin
- Re:拔掉网线后，原本的 TCP 连接还存在吗？  
还有一种情况，客户端明明断开了socket然后重新建立 tcp 连接，但是服务端在上一个连接收到了客户端新连接的数据，这个是什么原因呢？  
——多梦的歲月