**暨南大学本科实验报告专用纸**

课程名称 C++程序设计 成绩评定

实验项目名称 重载和引用函数编程 指导教师 王勇杰

实验项目编号 13 实验项目类型 编程型 实验地点 C105

学生姓名 李炜鹏 学号 2017052544

学院 智能科学与工程学院 系 计算机 专业 信息安全

实验时间2019年12月10日 午～12月13日 午 温度 ℃湿度

1. **实验目的**
2. 理解并且熟悉重载和引用函数
3. 惹事并理解对象移动
4. 理解左值与右值的区别
5. 实验要求：阅读课本第13章的内容， 试调试运行实验十三中的源代码
6. 实验报告要求：
   1. 实验目的
   2. 实验原理
   3. 注释必要的代码
   4. 实验结果的截图
   5. 实验评估
7. **实验原理**
8. 右值与左值：
   1. 当一个对象被用作右值时，实际使用的是对象的值（内容）；当一个对象被用作左值时，用的是对象的身份（在内存中的位置）
   2. 左值持久，右值短暂：左值有持久的状态（例如本实验中第二个sorted返回的ret，是函数体内对象的一个副本，有自己的内存空间，是持久的）；右值要么是字面变量，要么是在表达式求值过程中创建的临时变量（如本实验中第一个sorted返回的是临时创建的一个Foo类对象）
   3. 左值引用：不能将其绑定到要求转换的表达式、字面常量或是返回右值的表达式

右值引用：所引用的对象将要被销毁、该对象没有其他用户

1. **主要仪器设备**

**仪器：**计算机

**实验环境：**Visual Studio 2017或Dev-C++

1. **源程序**

**Lab13\_2.cpp**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

class Foo {

public:

Foo sorted() && ;//可用于可改变的右值

Foo sorted() const &;//可用于任何类型的Foo

Foo sorted(int);

private:

vector<int> data;

};

//本对象为右值，因此可以原址排序

Foo Foo::sorted() &&

{

cout << "Rvalue reference version." << endl;

sort(data.begin(), data.end());//排序

return \*this;

}

//本对象是const或是一个左值，那种情况我们都不能对其进行原址排序

Foo Foo::sorted() const &

{

cout << "Lvalue reference version." << endl;

Foo ret(\*this);//拷贝一个副本

sort(ret.data.begin(), ret.data.end());//排序副本

return ret;//返回副本

}

Foo retVal()

{

return Foo();//返回一个右值，是一个临时变量

}

Foo& retFoo()

{

return \*(new Foo());//返回一个左值

}

int main(void)

{

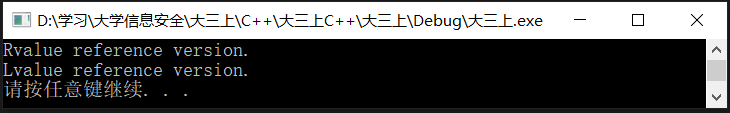
retVal().sorted();

retFoo().sorted();

system("pause");

return 0;

}



本程序定义了一个Foo类类体有三个移动成员函数，这里讨论前两个的区别：第一个将函数生命为返回右值引用，可用于可改变的右值，因此可以原址排序；第二个将函数声明为返回Foo类的const引用，由于返回的是Foo对象的一个副本，哪种情况下我们都不能对其进行原址排序。实验中分别调用了函数retVal和retFoo，分别产生了一个右值和一个左值，并且调用类的成员函数sorted，实验结果符合预期。

1. **实验评估**

通过本实验，我深刻地理解到了右值与左值的区别，认识到右值对象可以原址排序和左值对象则不能的原理（由于实验时间有限，并未深入了解，留作以后深入研究），通过上机实验，体会到了函数重载与引用的概念及意义。