

深圳大学实验报告

课程名称： 面向对象程序设计

实验项目名称： 实验八 运算符重载

学院： 医学院

专业： 生物医学工程

指导教师： 李乔亮、邓云

报告人： 陈焕鑫 学号： 2016222042 班级： 生工2班

实验时间： 2018.12.19

实验报告提交时间： 2018.12.20

教务部制

实验目的:

熟练掌握 C++ 中运算符重载的功能

实验内容:

完成字符串 CMyString 类的编写:

- 1、提供正确的无参、有参及拷贝构造函数、析构函数、打印输出函数
- 2、进行以下运算符的重载+、=、+=
- 3、在主函数中，对上述操作进行测试，验证其正确性

-----示例程序-----

```
CMyString a("abc"),b("def"),c
```

```
c = a + b;
```

```
a += b;
```

注 1: +表示字符串连接操作, "abc"+"def" = "abcdef"

注 2: 通过对结果的打印输出, 观察+与+=操作的不同

注 3: 必须提供正确的拷贝构造函数, 否则程序会出错, 思考为什么?

实验环境与程序代码：

实验环境：win10 系统下的 Visual Studio 2017

程序代码如下所示：

本程序包括 main.cpp, MyString.h 和 MyString.cpp 三个文件。MyString.h 文件中对 CMyString 类作了声明，在 CMyString 类中有私有变量：字符类型的指针 mystr，int 型 length。公有函数：无参构造函数、有参构造函数、拷贝构造函数、析构函数、“=” “+” “+=” 运算符的重载函数和一个打印函数。在 MyString.cpp 中对几个函数进行了实现。无参构造函数默认使 mystr 为空指针，length 为 0，代表没有接收到字符串。其他情况下首先调用 strlen 函数获得传入的字符串的长度，再将 length 进行相应的修改，使它符合新的字符串所具有的长度，然后使用 new 向系统申请一片足够大的内存区（大小为 length+1，多出来的一位存在字符串结束符“\0”）。让对象的字符串指针指向这片区域，如果该区域不为空（NULL），那么就调用 strcpy 拷贝字符串或者 strcat 拼接字符串。打印函数是直接将字符串使用 cout 打印出来。

```
//MyString.h
#ifndef _MY_STRING_H_
#define _MY_STRING_H_

class CMyString
{
private:
    char *mystr;    //字符串指针
    int length;     //字符串长度
public:
    CMyString();    //无参构造函数
    CMyString(const char* i_string); //有参构造函数
    CMyString(const CMyString &copy); //拷贝构造函数
    ~CMyString();   //析构函数
    CMyString operator= (const CMyString &operand); //重载运算符“=”
    CMyString operator+ (const CMyString &operand); //重载运算符“+”
    CMyString operator+= (const CMyString &operand); //重载运算符“+=”
    void printStr(); //打印
};

#endif

//MyString.cpp
#include <iostream>
#include "MyString.h" //包含对应的头文件
using namespace std; //使用命名空间 std

CMyString::CMyString() //无参构造函数
```

```

{
    mystr = NULL;           //默认字符串为空
    length = 0;             //默认长度为0
}

CMyString::CMyString(const char* i_string) //有参构造函数
{
    length = strlen(i_string); //长度为传入字符串的长度
    mystr = new char[length + 1]; //申请足够的内存空间

    if (mystr)                 //如果申请成功
    {
        strcpy(mystr, i_string); //调用拷贝字符串函数
    }
}

CMyString::CMyString(const CMyString &copy) //拷贝构造函数
{
    length = strlen(copy.mystr); //长度为拷贝对象的字符串的长度
    mystr = new char[length + 1]; //申请足够的内存空间

    if (mystr)                 //如果申请成功
    {
        strcpy(mystr, copy.mystr); //调用拷贝字符串函数
    }
}

CMyString::~CMyString() //析构函数
{
    delete[] mystr; //释放内存空间
}

CMyString CMyString::operator= (const CMyString &operand) //重载运算符"="
{
    length = strlen(operand.mystr); //长度为赋值对象的字符串的长度
    mystr = new char[length + 1]; //申请足够的内存空间

    if (mystr)                 //如果申请成功
    {
        strcpy(mystr, operand.mystr); //调用拷贝字符串函数
    }

    return *this; //返回自身
}

```

```

CMyString CMyString::operator+ (const CMyString &operand) //重载运算符"+"
{
    CMyString tmp;        //用来存放相加的结果

    tmp.length = this->length + operand.length; //长度为原长度加上另一个对象的字符串的长度的值
    tmp.mystr = new char[tmp.length + 1];        //申请足够的内存空间

    if (tmp.mystr)                                //如果申请成功
    {
        strcpy(tmp.mystr, this->mystr);           //先拷贝
        strcat(tmp.mystr, operand.mystr);         //再拼接
    }

    return tmp;        //返回相加的结果
}

CMyString CMyString::operator+= (const CMyString &operand) //重载运算符"+="
{
    CMyString tmp(*this); //拷贝构造一个临时对象
    delete[]mystr;         //释放原来的内存空间
    length += strlen(operand.mystr); //长度为原长度加上另一个对象的字符串的长度的值
    mystr = new char[length + 1];    //申请足够的内存空间

    if (mystr)                //如果申请成功
    {
        strcpy(mystr, tmp.mystr); //先拷贝
        strcat(mystr, operand.mystr); //再拼接
    }

    return *this;        //返回自身
}

void CMyString::printStr()
{
    cout << mystr; //打印
}

//main.cpp
#include <iostream>
#include "MyString.h"
using namespace std; //使用命名空间 std

```

```

int main()
{
    CMyString a("abc"), b("def"), c; //声明对象
    c = a + b;    //使用符号"+"和"="
    a += b;       //使用符号"+="

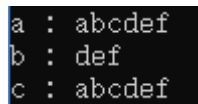
    cout << "a : ";
    a.printStr(); //打印字符串 a
    cout << endl << "b : ";
    b.printStr(); //打印字符串 b
    cout << endl << "c : ";
    c.printStr(); //打印字符串 c
    cout << endl;

    return 0;
}

```

实验结果与分析:

程序运行结果如图所示:



```

a : abcdef
b : def
c : abcdef

```

a 的初始值为 abc, b 的初始值为 def, c 为字符串 a+字符串 b 的结果, 即 abcdef, 之后又对 a 进行了+=b 的操作, 使得 a 自身加上 def, 成为 abcdef, 由图片可以看出程序运行结果正确。

从函数实现的角度来看, 重载 “+” 函数中, 并未改变原对象的值, 而是在函数中新声明了一个对象, 将两个旧的对象相加的结果存放在新的对象中, 将该新对象作为返回值从函数中返回, 而重载 “+=” 函数会修改原对象的值, 两个对象相加之后的值会赋给前者, 之所以在函数中又声明了一个新的对象, 并拷贝原对象的内容, 是因为原对象要进行拼接, 势必会改变长度, 需要重新申请一片新的内存, 所以先用新对象来存放原来的内容。

必须提供拷贝构造函数是因为重载运算符函数会返回 CMyString 类的结果, 在 return 的时候, 就需要调用拷贝构造函数来生成一个 CMyString 类的临时变量, 用来作为返回值。如果没有写拷贝构造函数, 程序无法编译通过。

指导教师批阅意见：

成绩评定：

指导教师签字：

年 月 日

备注：

- 注： 1、报告内的项目或内容设置，可根据实际情况加以调整和补充。
2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。