

CS205 C/ C++ Programming - Lab Assignment 5

Name: Jiachen Zhang

SID: 11713020

Part 1 - Analysis

This Lab Assignment focuses on the overload of the functions, and I need to learn about it.

I need to learn about the friend functions.

I finish this Lab Assignment with the following small steps.

- Basic procedures are similar to what I did in last assignment.
- Then I finish the four methods, which are\
 - `friend UTF8string operator+(const UTF8string str1, const UTF8string str2)` , that will become regular concatenation (if two objects are called str1 and str2, str1 + str2 changes neither str1 nor str2)\
 - `UTF8string operator+=(const UTF8string str)` , that is to append something (str1 += str2 changes str1, not str2)\
 - `friend UTF8string operator*(int times, const UTF8string str)` and `friend UTF8string operator*(const UTF8string str, int times)` , for repeating a string n times (if u is "àéeç", u * 2 or 2 * u should return "àéeçàéeç", u unchanged)
 - `friend UTF8string operator!(const UTF8string str)` , for reversing a string (original string unmodified), which means reversing the characters (not the bytes!), for instance if u is "étudiant" (student in French), !u should be "tnaiduté".

The thoughts and comments are in **Part 2 - Code**.

Part 2 - Code

```

#ifndef UTF8STRING_HPP
#define UTF8STRING_HPP

#include <iostream>
#include <string>
#include "utf8.h"

// THIS FILE DOESN'T NEED TO BE MODIFIED
// YOU NEED TO WRITE UTF8string.cpp

class UTF8string
{
    std::string m_s;

public:
    // Constructors - No code required
    UTF8string(){};
    UTF8string(const char *s) { m_s = std::string(s); };
    UTF8string(std::string s) { m_s = s; };
    // -----
    // value() is for checking what was done -> No code required
    std::string value() { return m_s; };

public:
    friend UTF8string operator+(const UTF8string str1, const UTF8string str2)
    {
        /* str1 and str2 will not be changed */
        UTF8string result = UTF8string(str1.m_s + str2.m_s);
        return result;
    }

    UTF8string operator+=(const UTF8string str)
    {
        /* str will not be changed
        while the variable on the left of '+= ' will be changed */
        this->m_s += str.m_s;
        return *this;
    }

    friend UTF8string operator*(int times, const UTF8string str)
    {
        if (times < 1)
            return "";
        UTF8string result;
        /* repeat and append for times times */
        while (times-->0)
            result += str;
        return result;
    }

    friend UTF8string operator*(const UTF8string str, int times) {
        /* for different order */
        return times * str;
    }

    friend UTF8string operator!(const UTF8string str)
    {
        std::string result = "";
        unsigned char *p = (unsigned char *) (str.m_s.c_str());
        int begin = 0;

```

```

    int lenptr = 0;
    while (*p != '\0' && *p != '\n' && *p != '\r')
    {
        lenptr = isutf8(p);
        p += lenptr;
        /* append to the head of the string */
        result = str.m_s.substr(begin, lenptr).append(result);
        begin += lenptr;
    }
    return UTF8string(result);
}
};
#endif

```

Part 3 - Result & Verification

I use testUTF8string.cpp to test my UTF8string.cpp .

```

#define __FileName testUTF8string.cpp
#include <iostream>
#include "UTF8string.hpp"

using namespace std;
int main(int argc, char **argv)
{
    cout << "-----" << endl;
    UTF8string str1 = UTF8string("123");
    UTF8string str2 = UTF8string("456");
    UTF8string str3 = UTF8string("1张2佳3晨");
    UTF8string result;
    cout << "str1: " << str1.value() << endl;
    cout << "str2: " << str2.value() << endl;
    cout << "str3: " << str3.value() << endl;
    cout << "-----" << endl;
    result = str1 + str2;
    cout << "str1 + str2: " << result.value() << endl;
    cout << "str1: " << str1.value() << endl;
    cout << "str2: " << str2.value() << endl;
    cout << "-----" << endl;
    str2 += str1;
    cout << "str2 += str1, str2: " << str2.value() << endl;
    cout << "-----" << endl;
    result = 3 * str1;
    cout << "3 * str1: " << result.value() << endl;
    result = str1 * 2;
    cout << "str1 * 2: " << result.value() << endl;
    cout << "-----" << endl;
    result = !str3;
    cout << "!str3: " << result.value() << endl;
    cout << "-----" << endl;
    return 0;
}

```

Input

Output:

```
-----
str1: 123
str2: 456
str3: 1张2佳3晨
-----
str1 + str2: 123456
str1: 123
str2: 456
-----
str2 += str1, str2: 456123
-----
3 * str1: 123123123
str1 * 2: 123123
-----
!str3: 晨3佳2张1
-----
```

Part 4 - Difficulties & Solutions

friend function is interesting that I can not discribe it clearly but get some idea of it.

The overload of the operator methonds is a new thing for me, and I have learned about some related to overloading and operator method after finish this assignment.

For developing good programming habits, I use `std::cout`, `std::endl` and `std::string`, etc. instead using `using namespace std;` in the `UTF8string.hpp` file.