3460:209-010 Fall 2011 Lab 4 Report

Name:
UANET id: ig11

Generated: Fri Oct 23 10:25:43 EDT 2015

```
/home/research/hvn1/cs210fa15/010/students/ig11/Labs/Lab4/main.cpp:26:1: error: stray â\200\230‘â\200\231 in program
 b = "three";‘
 ^
/usr/bin/timeout: failed to run command â\200\230/home/research/hvn1/cs210fa15/010/students/ig11/Labs/Lab4/runlabâ\200\231: No such file or directory
```

Note: First column is revision number when that line was last changed.

```
------------ ig11/Labs/Lab4/string.cpp ------------
/********************************************************************
  * Lab 3: Make Your Own String Class string.cpp
  * Author: Istvan Gates
  *         ig11@zips.uakron.edu
  *
  * Purpose: make my own string class with constructor and deconstructor
  *
  *
  ********************************************************************/
#include <iostream>
#include "string.hpp"



------------ ig11/Labs/Lab4/CMakeLists.txt ------------

        project(Lab4  CXX)
cmake_minimum_required(VERSION 2.8.11)

set(CMAKE_CXX_FLAGS "-std=c++11")

add_executable(lab4
        string.cpp
        main.cpp)

------------ ig11/Labs/Lab4/string.hpp ------------
#ifndef STRING_HPP
#define STRING_HPP
#include <string.h>
#include <iostream>
//create String class
class String{
        public:
//declare member variables and functions
                int len;
                char* str;

                //default constructor for String
                String()
                        :len(0), str(new char[1])
                {}

                //copy constructor for String. copies the length of existing char array, dynamically allocates memory for the new string
                String(String const& s)
                        :len(s.len), str(new char[len+1])
                {
                        strcpy(str, s.str);
                }

                //c-string constructor. copies length and contents, then dynamically allocates memory for new c string array
                String(char const* s)
                        :len(strlen(s)), str(new char[len +1])
                {
                        strcpy(str, s);
                }

                //destructor to free up dynamically allocated memory
                ˜ String(){
                        delete[] str;
                }

                //function to get the member varialbe

                char const* getString()const{
                        return str;
                }
//perform copy operation
                String& operator=(char const* s){
                        len = strlen(s);
                        delete[] str;
                        str = new char[len + 1];

                        strcpy(str, s);
                        return *this;
}
//perform copy operation
                String& operator=(String const& s){
                        len = s.len;
                        delete[] str;
                        str = new char[len + 1];
                        strcpy(str, s.str);
                        return *this;
                }
};
//ovleroading the cin operator
inline std::ostream& operator<<(std::ostream& os, String const& s)
{
        return os << s.getString();
}
//overlading the + operator so that concatenation may be performed
inline String operator+(String const& lhs, String const& rhs) {
        String result;
        result.len = (lhs.len + rhs.len);
        result.str = new char[result.len + 1];
        strcpy(result.str, lhs.str);
        return result;
}
//overloading all of the operators
//starting with == operator, using strcmp
inline bool operator ==(String const& lhs, String const& rhs){
        int rc = strcmp(lhs.str, rhs.str);
        if(rc == 0)
                return true;
        else
                return false;
}
//perform != overloading, by just notting the == operatio and notting it
inline bool operator!=(String const& lhs, String const& rhs){
        return !(lhs == rhs);
}
//using strcmp to see if strings are greater than
inline   bool operator >(String const& lhs, String const& rhs){
        int rc = strcmp(lhs.str, rhs.str);
        if(rc == 1)
                return true;
        else
                return false;
}
//once again check to see if strings are equal or greater than
inline   bool operator >=(String const& lhs, String const& rhs){
        int rc = strcmp(lhs.str, rhs.str);
        if(rc == 0 || rc == 1)
                return true;
        else return false;
}

//check to see if they are less than, see if result of strcmp is less than 0
inline   bool operator <(String const& lhs, String const& rhs){
        int rc = strcmp(lhs.str, rhs.str);
        return !(lhs > rhs);
}
//check to see if result of strcmp is 0 or less than 0
inline   bool operator <=(String const& lhs, String const& rhs){
        int rc = strcmp(lhs.str, rhs.str);
        if(rc == 0 || rc == -1)
                return true;
        else
                return false;
}
//end the header guard :)
#endif

------------ ig11/Labs/Lab4/CMakeList.txt ------------
project(Lab4 CXX)
cmake_minimum_required(VERSION 2.8.11)

set(CMAKE_CXX_FLAGS "-std=c++11")

add_executable(lab4
        string.cpp
        main.cpp)

------------ ig11/Labs/Lab4/main.cpp ------------
/*******************************************************************
  * Lab 3: Make Your Own String Class main.cpp
  *
  * Author: Istvan Gates
  *         ig11@zips.uakron.edu
  *
  * Purpose: make my own string class with constructor and deconstructor
  *
  ********************************************************************/
#include "string.hpp"
#include <cassert>

int main(){
//initialize all of my variables, and perform the deep copy operation
  String s1;
  String s2 = "test";
  String s3 = s2;

  std::cout << s1 << ’\n’;
  std::cout << s2 << ’\n’;
  std::cout << s3 << ’\n’;
        //perform reassignment
String a = "one";
String b = "two";
a = b;
b = "three";‘
std::cout << a << ’\n’;
std::cout << b << ’\n’;
//perform concatenation operation
String hello  = "hello ";
String world = "world";
String concat = hello + world;
std::cout << concat << ’\n’;
String c = "c";
String d = "d";
//assert tests
assert(c == c);
assert(c != d);
assert(c < d);
assert(c <= d);
assert(d > c);
assert(d >= c);

}

------------------------------------------------------------------------
r848 | ig11 | 2015-10-13 18:20:32 -0400 (Tue, 13 Oct 2015) | 1 line

my last commit
------------------------------------------------------------------------
r840 | ig11 | 2015-10-13 16:08:27 -0400 (Tue, 13 Oct 2015) | 1 line

final commit
------------------------------------------------------------------------
r839 | ig11 | 2015-10-13 16:08:21 -0400 (Tue, 13 Oct 2015) | 1 line

final commit
------------------------------------------------------------------------
r838 | ig11 | 2015-10-13 16:08:12 -0400 (Tue, 13 Oct 2015) | 1 line

final commit
------------------------------------------------------------------------
r837 | ig11 | 2015-10-13 16:07:59 -0400 (Tue, 13 Oct 2015) | 1 line

final commit
------------------------------------------------------------------------
r796 | ig11 | 2015-10-12 16:45:08 -0400 (Mon, 12 Oct 2015) | 1 line

my first commit
------------------------------------------------------------------------
r494 | hvn1 | 2015-10-04 17:15:13 -0400 (Sun, 04 Oct 2015) | 1 line

added lab4 folders for bot sections
------------------------------------------------------------------------
```