

C:\Users\Zachi\Dropbox\zCpp\Teach_2014\CopyandAssignment\Debug\CopyandAssignment.exe

```
headline2: h2: Lettuce Prey
sports: Spinach leaves bowl

String pass
  "h1: Cel
headline1:


String pass
  "h2: Let
'h2: Lettuc
num_strings
headline2:

Initialize
sailor: "Sp

Assign one
num_strings
knot: "Spin

End of main
Press any k
"Spinach le
num_strings
"€|€|€|€|€|
|€|€|€|€|€|
|€|€|€|€|€|
|€|€|€|€|€|
default deleted
num_strings = 1 left.
```

Microsoft Visual C++ Runtime Library

 Debug Assertion Failed!

Program:
...zCpp\Teach_2014\CopyandAssignment\Debug\CopyandAssignment.exe
File: f:\dd\vctools\crt_bld\self_x86\crt\src\dbgdel.cpp
Line: 52

Expression: _BLOCK_TYPE_IS_VALID(pHead->nBlockUse)

For information on how your program can cause an assertion failure, see the Visual C++ documentation on asserts.

(Press Retry to debug the application)

Abort Retry Ignore

€|€|€|€|€|
€	€	€	€	€
€	€	€	€	€
€	€	€	€	€

The right solution:

```
ca. C:\windows\system32\cmd.exe
num_strings=1 : "h1: Celery Stalks at midnight" object created
num_strings=2 : "h2: Lettuce Prey" object created
num_strings=3 : "Spinach leaves bowl" object created
headline1: h1: Celery Stalks at midnight
headline2: h2: Lettuce Prey
sports: Spinach leaves bowl

String passed by reference:
  "h1: Celery Stalks at midnight"
headline1: h1: Celery Stalks at midnight

num_strings=4 : "h2: Lettuce Prey" Copy object created
String passed by value:
  "h2: Lettuce Prey"
"h2: Lettuce Prey" default deleted
num_strings = 3 left.
headline2: h2: Lettuce Prey

Initialize one object to another:
num_strings=4 : "Spinach leaves bowl" Copy object created
sailor: "Spinach leaves bowl"

Assign one object to another:
num_strings=5 : "C++" default object created
num_strings=5 : "Spinach leaves bowl" Assignment object
knot: "Spinach leaves bowl"

End of main()
Press any key to continue . . .
"Spinach leaves bowl" default deleted
num_strings = 4 left.
"Spinach leaves bowl" default deleted
num_strings = 3 left.
"Spinach leaves bowl" default deleted
num_strings = 2 left.
"h2: Lettuce Prey" default deleted
num_strings = 1 left.
"h1: Celery Stalks at midnight" default deleted
num_strings = 0 left.
Press any key to continue . . . _
```

```

// String1.h

#ifndef _STRING1_H_
#define _STRING1_H_

#include <iostream>

class String1
{
private:
    // ED-NOTE: using pointer, not array. That means we will need to
    // use "new" to allocate memory
    char *str;
    int len;                // length

    // ED-Note: Using static variable. ONE for all objects.
    static int num_strings; // number of string objects

public:
    String1(const char *s);
    // ONLY add as the remedy!!
    //String1(const String1& s);
    // ONLY add as the remedy!!
    //String1 & String1::operator=(const String1& s);

    String1();
    ~String1();

    friend std::ostream & operator<<(std::ostream & os,
        const String1 & st);
};

#endif

```

```

// String1.cpp

#include <cstring>
#include "String1.h"

using std::cout ;

// Static member
int String1::num_strings = 0;

// Class Methods

// Constructor: from C string
String1::String1(const char *s)
{
    len = std::strlen(s);    // strlen() returns W/O the trailing '\0'
    str = new char[len+1];
    std::strcpy(str,s);
    num_strings++;
    cout << "num_strings=" << num_strings << " : \n" <<
        str << "\n object created\n";
}

// Constructor: default
String1::String1()
{
    len = 4;
    str = new char[4];
    std::strcpy(str,"C++");    // Default string
    num_strings++;
    cout << "num_strings=" << num_strings << " : \n" <<
        str << "\n default object created\n";
}

/*
// Copy constructor (only add later!)
String1::String1(const String1& s)
{
    len = s.len;
    str = new char[len+1];
    std::strcpy(str,s.str);
    num_strings++;
    cout << "num_strings=" << num_strings << " : \n" <<
        str << "\n Copy object created\n";
}

// Assignment operator (only add later!)
String1 & String1::operator=(const String1& s)
{
    if (this == &s)
        return *this;
    delete [] str;
    len = s.len;
    str = new char[len+1];
    std::strcpy(str,s.str);
    //num_strings++;
    cout << "num_strings=" << num_strings << " : \n" <<
        str << "\n Assignment object \n";
    return *this;
}

```

```
}
*/
// Destructor
String1::~String1()
{
    cout << "\"" <<
        str << "\" default deleted\n";
    --num_strings;
    cout << "num_strings = " << num_strings << " left.\n";
    delete [] str;
}

std::ostream & operator<<(std::ostream & os, const String1 & st)
{
    os << st.str;
    return os;
}
```

```

// main.cpp
// Test file for String1.cpp

#include <iostream>
using std::cout;
using std::endl;

#include "String1.h"

void callmebyref(String1 &);
void callmebyval(String1);

int main()
{
    String1 headline1("h1: Celery Stalks at midnight");
    String1 headline2("h2: Lettuce Prey");
    String1 sports("Spinach leaves bowl");

    cout << "headline1: " << headline1 << endl;
    cout << "headline2: " << headline2 << endl;
    cout << "sports: " << sports << endl;
    cout << "\n\n";
    callmebyref(headline1);
    cout << "headline1: " << headline1 << endl;

    cout << "\n\n";
    callmebyval(headline2);
    cout << "headline2: " << headline2 << endl;

    cout << "\n\n";
    cout << "Initialize one object to another:\n" ;
    String1 sailor = sports;
    cout << "sailor: \"" << sailor << "\"" << endl;

    cout << "\n\n";
    cout << "Assign one object to another:\n";
    String1 knot;
    knot = sports;
    cout << "knot: \"" << knot << "\"" << endl;

    cout << "\n\n";
    cout << "End of main()\n" ;
    system("pause");
    // Method 2:
    // Project -> Properties -> Configuration Properties -> Linker -> System.
    // Then select Console (/SUBSYSTEM:CONSOLE) in "SubSystem" option.
    // AND, when you run, run in non-debug mode (CTRL-F5)

    return 0;
}

```

```
void callmebyref(String1 & s)
{
    cout << "String passed by reference:\n";
    cout << "  \"" << s << "\"\n";
}
```

```
void callmebyval(String1 s)
{
    cout << "String passed by value:\n";
    cout << "  \"" << s << "\"\n";
}
```