STRINGS (THE STRING TABLE)

Exercise Files

Starter – "engine/exercises/chapter10/exer_010.cc"
Answers – "engine/answers/chapter10/exer_010.cc"

Exercise Mission

n/a

Special Steps

Please remember, when you modify the engine and compile, you must copy the new executable over to your Kit/ directory before you can run it and see the changes in the Kit (as instructed below).

Synopsis

In this exercise, we will examine some different StringTable operations and test your knowledge of how they should work.

Prerequisites

1. *ch1_001.pdf "Using The Kit"*

Exercises

1. *String Table Tests (pg 2)*

STRINGS (THE STRING TABLE)

1 String Table Tests

Goal: Identify the correct behavior of various String Table operations.

Starter Code: A console function definition is provided for this exercise (ch10_exer_010()).

In the first part of the implementation, three unique strings are provided.

```
ConsoleFunction(ch10_exer_010, void, 1, 1, "")
{
   const char * string1 = "Hello World!";
   const char * string2 = "Hello World! IGNORE THIS";
   const char * string3 = "HELLO WORLD!";
```

A bit further in, we these strings are added to the string table using three different operations.

```
   // Add the above const char * strings in different ways
   StringTable->insert(string1);
   StringTable->insertn(string2, 12);
   StringTable->insert(string3, true);
```

Then, nine different tests are run on the string table.

```
   // Do some tests
   // 1
   myString1 = StringTable->lookup( string1 );
   Con::printf("Test 1: string1 found? %s",
               (myString1) ? "YES" : "NO" );

   // 2
   myString1 = StringTable->lookup( string2 );
   Con::printf("Test 2: string2 found? %s",
               (myString1) ? "YES" : "NO" );

   // 3
   myString1 = StringTable->lookup( string3 );
   Con::printf("Test 3: string1 found? %s",
               (myString1) ? "YES" : "NO" );

   // 4
   myString1 = StringTable->lookup( string1 );
   Con::printf("Test 4: Same string? %s",
               (myString1 == string1) ? "YES" : "NO" );

   // 5
   myString1 = StringTable->lookup( string3 );
   Con::printf("Test 5: Same string? %s",
               (myString1 == string3) ? "YES" : "NO" );
```


STRINGS (THE STRING TABLE)

```
// 6
myString1 = StringTable->lookup( string1 );
myString2 = StringTable->lookup( string3 );
Con::printf("Test 6: Same string? %s",
            (myString1 == myString2) ? "YES" : "NO" );

// 7
myString1 = StringTable->lookup( string1 );
myString2 = StringTable->lookup( string3 , true );
Con::printf("Test 7: Same string? %s",
            (myString1 == myString2) ? "YES" : "NO" );

// 8
hash1 = StringTable->hashString( string1 );
hash2 = StringTable->hashString( string2 );
Con::printf("Test 8: Same hash? %s",
             (hash1 == hash2) ? "YES" : "NO" );

// 9
hash1 = StringTable->hashString( string1 );
hash2 = StringTable->hashStringn( string2 , 12);
Con::printf("Test 9: Same hash? %s",
             (hash1 == hash2) ? "YES" : "NO" );
```

The tests are,

- Test 1 – Ignoring case, check to see if "string1" is in the StringTable.
- Test 2 – Ignoring case, check to see if "string2" is in the StringTable.
- Test 3 – Ignoring case, check to see if "string3" is in the StringTable.
- Test 4 – Check to see if "string1" has the same address as the string table entry for "string1".
- Test 5 – Check to see if "string3" has the same address as the string table entry for "string3".
- Test 6 – Ignoring case, check to see if the values stored in the string table for "string1" and "string3" are the same.
- Test 7 – Not ignoring case, check to see if the values stored in the string table for "string1" and "string3" are the same.
- Test 8 – Ignoring case, check to see if "string1" and "string2" produce the same hash.
- Test 9 – Not ignoring case, check to see if "string1" and "string2" produce the same hash.

STRINGS (THE STRING TABLE)

Questions:

1. Before running ch10_exer_010(), for each of the tests, please circle the answer you think the test will produce:
   - Test 1 – YES / NO
   - Test 2 – YES / NO
   - Test 3 – YES / NO
   - Test 4 – YES / NO
   - Test 5 – YES / NO
   - Test 6 – YES / NO
   - Test 7 – YES / NO
   - Test 8 – YES / NO
   - Test 9 – YES / NO
2. What are the reasons for your answers?
   - Test 1 _____
   - Test 2 _____
   - Test 3 _____
   - Test 4 _____
   - Test 5 _____
   - Test 6 _____
   - Test 7 _____
   - Test 8 _____
   - Test 9 _____

