17.5 — std::string assignment and swapping

BY ALEX ON JULY 18TH, 2010 | LAST MODIFIED BY ALEX ON AUGUST 6TH, 2017

String assignment

The easiest way to assign a value to a string is to use the overloaded operator= function. There is also an assign() member function that duplicates some of this functionality.

string& string::operator= (const string& str)
string& string::assign (const string& str)
string& string::operator= (const char* str)
string& string::assign (const char* str)
string& string::operator= (char c)

- These functions assign values of various types to the string.
- These functions return *this so they can be “chained”.
- Note that there is no assign() function that takes a single char.

Sample code:

```
1  string sString;
2
3  // Assign a string value
4  sString = string("One");
5  cout << sString << endl;
6
7  const string sTwo("Two");
8  sString.assign(sTwo);
9  cout << sString << endl;
10
11 // Assign a C-style string
12 sString = "Three";
13 cout << sString << endl;
14
15 sString.assign("Four");
16 cout << sString << endl;
17
18 // Assign a char
19 sString = '5';
20 cout << sString << endl;
21
22 // Chain assignment
23 string sOther;
24 sString = sOther = "Six";
25 cout << sString << " " << sOther << endl;
```

Output:

```
One
Two
Three
Four
5
Six Six
```

The assign() member function also comes in a few other flavors:

string& string::assign (const string& str, size_type index, size_type len)

- Assigns a substring of str, starting from index, and of length len
- Throws an out_of_range exception if the index is out of bounds
- Returns *this so it can be “chained”.

Sample code:

```
const string sSource("abcdefg");
string sDest;

sDest.assign(sSource, 2, 4); // assign a substring of source from index 2 of length 4
cout << sDest << endl;
```

Output:

```
cdef
```

string& string::assign (const char* chars, size_type len)

- Assigns len characters from the C-style array chars
- Throws an length_error exception if the result exceeds the maximum number of characters
- Returns *this so it can be “chained”.

Sample code:

```
string sDest;

sDest.assign("abcdefg", 4);
cout << sDest << endl;
```

Output:

```
abcd
```

This function is potentially dangerous and its use is not recommended.

string& string::assign (size_type len, char c)

- Assigns len occurrences of the character c
- Throws a length_error exception if the result exceeds the maximum number of characters
- Returns *this so it can be “chained”.

Sample code:

```
string sDest;

sDest.assign(4, 'g');
cout << sDest << endl;
```

Output:

```
gggg
```

Swapping

If you have two strings and want to swap their values, there are two functions both named swap() that you can use.

void string::swap (string &str)
void swap (string &str1, string &str2)

- Both functions swap the value of the two strings. The member function swaps *this and str, the global function swaps str1 and str2.
- These functions are efficient and should be used instead of assignments to perform a string swap.

Sample code:

```
string sStr1("red");
string sStr2("blue");

cout << sStr1 << " " << sStr2 << endl;
swap(sStr1, sStr2);
cout << sStr1 << " " << sStr2 << endl;
sStr1.swap(sStr2);
cout << sStr1 << " " << sStr2 << endl;
```

Output:

```
red blue
blue red
red blue
```


17.6 -- std::string appending


Index


17.4 -- std::string character access and conversion to C-style arrays

Share this:

Facebook Twitter Google Pinterest

C++ TUTORIAL C++, PROGRAMMING, TUTORIAL | PRINT THIS POST

16 comments to 17.5 — std::string assignment and swapping

m4sk1n
August 5, 2017 at 7:37 am · Reply

“is to the use the”, probably “is to use the”…

apfelpektin
April 3, 2017 at 11:10 pm · Reply

quote from above:

```
string& string::assign (const char* chars, size_type len)
   [...]
   - Ignores special characters (including ”)
```

does anybody have some more information what that is about? it seems not to be the case here. double and single quotes for example are copied correctly from a string literal.