17.5 — std::string assignment and swapping

BY ALEX ON JULY 18TH, 2010 | LAST MODIFIED BY ALEX ON JANUARY 23RD, 2020

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:

```
string sString;

// Assign a string value
sString = string("One");
cout << sString << endl;

const string sTwo("Two");
sString.assign(sTwo);
cout << sString << endl;

// Assign a C-style string
sString = "Three";
cout << sString << endl;

sString.assign("Four");
cout << sString << endl;

// Assign a char
sString = '5';
cout << sString << endl;

// Chain assignment
string sOther;
sString = sOther = "Six";
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

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

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

awdawewwsd
May 4, 2019 at 3:35 pm · Reply

This is a question on chapter 15.
Instead of using std::weak_ptr, couldn't you use a normal pointer?