

22.5 — std::string assignment and swapping

ALEX AUGUST 26, 2021

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:

```
1 | const string sSource("abcdefg");
2 | string sDest;
3 |
4 | sDest.assign(sSource, 2, 4); // assign a substring of source from index 2 of length
  | 4
5 | cout << sDest << endl;
```

Output:

```
cdef
```

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

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

Sample code:

```
1 | string sDest;
2 |
3 | sDest.assign("abcdefg", 4);
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:

```
1 | string sDest;
2 |
3 | sDest.assign(4,
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 `s`
- These functions are efficient and should be used instead of assignments to perform a string swap.

Sample code:

```
1 | string sStr1("red");
2 | string sStr2("blue");
3 |
4 | cout << sStr1 << " " << sStr2 <<
   | endl;
5 | swap(sStr1, sStr2);
6 | cout << sStr1 << " " << sStr2 <<
   | endl;
7 | sStr1.swap(sStr2);
8 | cout << sStr1 << " " << sStr2 <<
   | endl;
```

Output:

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



Next lesson

22.6 `std::string` appending



Back to table of contents



Previous lesson

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

Leave a comment... Put C++ code between triple-backticks (markdown style):````Your C++ code`

 Name*

 Email*



Avatars from <https://gravatar.com/> are connected to your provided email address.

Notify me about replies:



POST COMMENT

DP N N FOUT

Newest ▼

