22.7 — std::string inserting

ALEX AUGUST 26, 2021

Inserting

Inserting characters into an existing string can be done via the insert() function.

string& string::insert (size_type index, const string& str)
string& string::insert (size_type index, const char* str)

- **Both functions insert the characters of str into the string at index**
- **Both function return *this so they can be "chained".**
- **Both functions throw out_of_range if index is invalid**
- **Both functions throw a length_error exception if the result exceeds the m**
- **In the C-style string version, str must not be NULL.**

Sample code:

```
string sString("aaaa");
cout << sString << endl;

sString.insert(2, string("bbbb"));
cout << sString << endl;

sString.insert(4, "cccc");
cout << sString << endl;
```

Output:

```
aaaa
aabbbbaa
aabbccccbbaa
```

Here's a crazy version of insert() that allows you to insert a substring into a string at an arbitrary index:

string& string::insert (size_type index, const string& str, size_type startindex, size_type num)

- This function inserts num characters str, starting from startindex, into the string at index.
- Returns *this so it can be "chained".
- Throws an out_of_range if index or startindex is out of bounds
- Throws a length_error exception if the result exceeds the maximum number of characters.

Sample code:

```
string sString("aaaa");

const string sInsert("01234567");
sString.insert(2, sInsert, 3, 4); // insert substring of sInsert from index [3,7) into sString at index 2
cout << sString << endl;
```

Output:

```
aa3456aa
```

There is a flavor of insert() that inserts the first portion of a C-style string:

string& string::insert(size_type index, const char* str, size_type len)

- Inserts len characters of str into the string at index
- Returns *this so it can be "chained".
- Throws an out_of_range exception if the index is invalid
- Throws a length_error exception if the result exceeds the maximum number of characters.
- Ignores special characters (such as ")

Sample code:

```
string sString("aaaa");

sString.insert(2, "bcdef", 3);
cout << sString << endl;
```

Output:

```
aabcdaa
```

There's also a flavor of insert() that inserts the same character multiple times:

string& string::insert(size_type index, size_type num, char c)

- **Inserts num instances of char c into the string at index**
- **Returns *this so it can be “chained”.**
- **Throws an out_of_range exception if the index is invalid**
- **Throws a length_error exception if the result exceeds the maximum number of characters.**

Sample code:

```
string sString("aaaa");

sString.insert(2, 4,
'c');
cout << sString <<
endl;
```

Output:

```
aaccccaa
```

And finally, the insert() function also has three different versions that use iterators:

void insert(iterator it, size_type num, char c)
iterator string::insert(iterator it, char c)
void string::insert(iterator it, InputIterator begin, InputIterator end)

- **The first function inserts num instances of the character c before the iterator it.**
- **The second inserts a single character c before the iterator it, and returns an iterator to the position of the character ins**
- **The third inserts all characters between [begin,end) before the iterator it.**
- **All functions throw a length_error exception if the result exceeds the maximum number of characters.**

Next lesson
23.1 Input and output (I/O) streams

Back to table of contents

Previous lesson
22.6 std::string appending

x

x