The Standard Template Library Applications

Lecture 36
Section 10.5

Robb T. Koether

Hampden-Sydney College

Wed, Apr 25, 2018

Outline

Outline

The stack Adaptor Class

- An adaptor class uses a container class.
- We may construct a stack in any of the following ways.

Ways to Construct a Stack

```
#include <stack>
int main()
{
    stack<int> s1;
    stack<int, vector<int>> s2;
    stack<int, deque<int>> s3;
    stack<int, list<int>> s4;
}
```

The stack Adaptor Class

- The `stack` class has the following member functions (besides the fundamental four).

`stack` Member Functions

```
bool empty() const;
int size() const;
T& top();
void push(const T& value);
void pop();
```

Outline

The map Container Class

- A map is an *associative* list.
- Each member has
  - A key.
  - A value.
- The key must be unique for that member.
- The value is accessed through the key, by matching the key.
- This sounds like a hash table.

The map Container Class

- Suppose we want to store a list of students and their declared majors.

Name	Major
John	Mathematics
Tim	Computer Science
Betty	Chemistry
Ann	Mathematics

The map Container Class

- If we intend to locate members by name, then
  - The name is the key
  - The major is the value.
- We construct the (empty) map:

Construct a `map`

```
#include <map>
map<string, string> major;
```

The map Container Class

- To add the data, we may use the subscript operator:

Initialize the `map`

```
major["Andy"] = "Mathematics";
major["Betty"] = "Computer Science";
major["Chuck"] = "Chemistry";
major["Debbie"] = "Mathematics";
```

The map Container Class

- To find `"John"`, we use the `find()` function.
- It returns an iterator to John's location in the map.

Search the `map`

```
map<string, string>::iterator it;
it = major.find("Andy");
```

The map Container Class

- The data members first and second store the key and the value.

Print the `map`

```
map<string, string>::iterator it;
for (it = major.begin(); it != major.end(); it++)
    cout << it->first << " is majoring in "
        << it->second << endl;
```

Outline

Assignment

Assignment

- Read Sections 9.7 - 9.8.