Programming Abstractions

CS106B

Cynthia Bailey Lee
Julie Zelenski

Stanford University

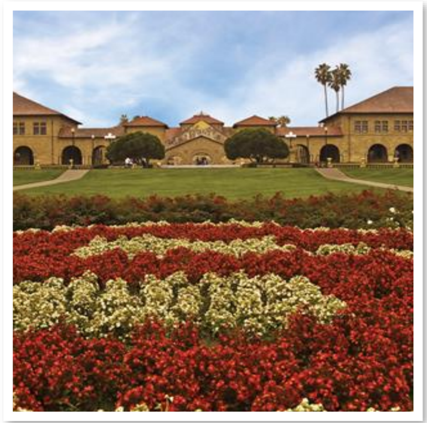
Today’s Topics

More ADTs!

- Map
  - Code example: counting words in text
- Containers-within-containers
  - Shallow copy vs. deep copy

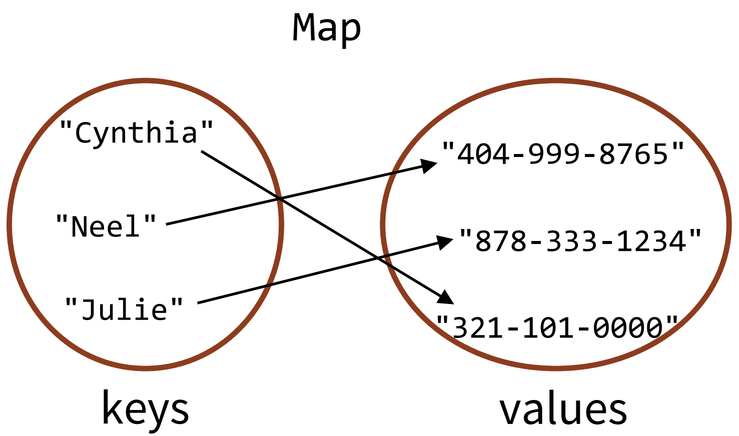
Maps

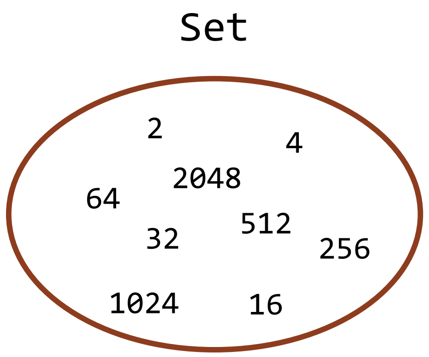
(not like the driving directions kind of maps though)

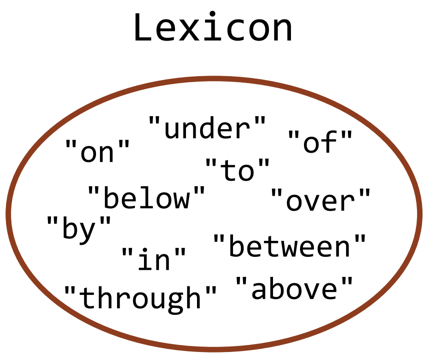


Associative containers

- Map
- Set
- Lexicon







Not as concerned with order but with association

- Map: associates **keys** with **values** (each could be any type)
- Set: associates **keys** with **membership** (in or out)
  - Lexicon: a set of strings, *with special internal optimizations for that*

Stanford University

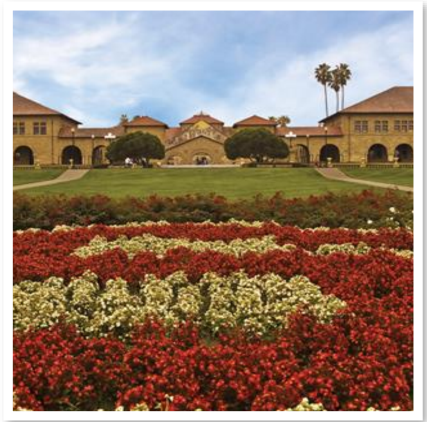
Stanford library Map *(selected member functions)*

```
void put(KeyType& key, ValueType& value);
bool containsKey(KeyType& key);
ValueType get(KeyType& key);
ValueType operator [](KeyType& key);
```

```
#include "map.h"

Map<string, string> phone;                // Map takes two(!) template parameters

phone["Cynthia"] = "321-101-0000";        // two syntax options for adding new item
phone.put("Julie", "878-333-1234");

if (phone.containsKey("Cynthia") && phone.containsKey("Julie")) {
    cout << phone["Cynthia"] << endl;     // two syntax options for getting item
    cout << phone.get("Julie") << endl;
    cout << phone["MTL"] << endl;         // what would this do??
}
```

Map Code Example

Tabulating word counts



Stanford University

Map programming exercise

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

- **First do an initial report:**
  - Print all words that appeared in the book at least 100 times, in alphabetical order
- **Then go into interactive query mode:**
  - The user types a word and we report *how many times* that word appeared in the book (repeat in a loop until quit).

Map programming exercise

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

- The user types a word and we report *how many times* that word appeared in the book (repeat in a loop until quit).

What would be a good design for this problem?

A. `Map<int, string> wordCounts;`

B. `Map<Vector<string>, Vector<int>> wordCounts;`

C. `Map<Vector<int>, Vector<string>> wordCounts;`

D. `Map<string, int>  wordCounts;`

E. `Map<string, Vector<int>> wordCounts;`

F. Other/none/more

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

How can we record the count?

(In other words, what goes in the place marked "record count here" in the code at right?)

A. `wordCounts[word] += word;`
B. `wordCounts[word] += 1;`
C. `wordCounts[word]++;`
D. B and C are good, but you need to first detect new (never seen before) words so you can start at zero before you start adding +1
E. Other/none/more

```
// We are given a vector that is just the
// the book, broken into pieces based on
// spaces between words. The type is:
// Vector<string> words;

Map<string, int> wordCounts;
for (string word : words) {
    // record count here
}
```

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

- The user types a word and we report *how many times* that word appeared in the book (repeat in a loop until quit).

```
// userWord is a word the user typed into the console
cout << userWord << " appears " << wordCounts[userWord] << " times" << endl;
```

What happens if queryWord is not a word in the book? 🤔

- Will the program crash?
- What other issue(s) besides crash do you foresee?

Write a program to count the number of occurrences of each unique word in a text file (e.g. *Poker* by Zora Neale Hurston).

- Report all words that appeared in the book at least 100 times, in alphabetical order

```
for (string word : wordCounts) {
   if (wordCounts[word] >= FREQUENCY_THRESHOLD) {
      cout << word << "\t" << wordCounts[word] << endl;
   }
}
```

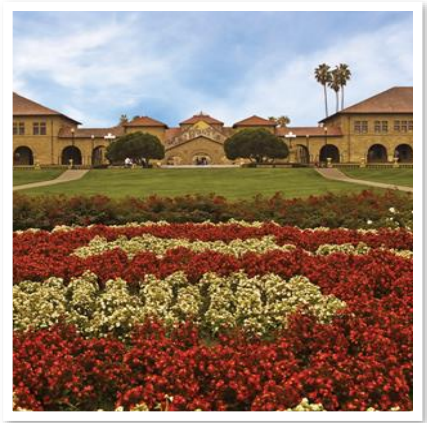
Does this work for our alphabetical order requirement?

- Yes!
- Stanford library Map returns its keys in sorted order

Compound Containers

Containers within containers within containers!

It’s turtles all the way down...



Can we add the number 4 to a Vector? Let's see…

```
Vector<int> numbers;
numbers.add(1);
numbers.add(2);
numbers.add(3);
Map<string, Vector<int>> mymap;
mymap["abc"] = numbers;
// Now we want to add 4 to the Vector inside the Map, how can we do it?
```

```
numbers.add(4);
```

```
mymap["abc"].add(4);
```

```
Vector<int> test = mymap["abc"];
test.add(4);
```

Would any of these three options would work if inserted here? Which one(s)? Why or why not?

```
// GOAL: we want this to print 4 (indicating the add(4) worked)
cout << "New size: " << mymap["abc"].size() << endl;
```

Can we add the number 4 to a Vector? Let’s see…

You don’t need to worry too much about the details of how the cases differ in terms of behind-the-scenes mechanism—I just wanted to flag it as a potential issue in case you accidentally encounter this in your code!

```
> mymap;

to the Vector inside the Map, how can we do it?
```

```
numbers.add(4);
```

```
mymap["abc"].add(4);
```

```
Vector<int> test = mymap["abc"];
test.add(4);
```

Would any of these three options would work if inserted here? Which one(s)? Why or why not?

```
// GOAL: we want this to print 4 (indicating the add(4) worked)
cout << "New size: " << mymap["abc"].size() << endl;
```

Stanford University