STL

C++'s Standard Containers Library

Matthew Barulic

RoboJackets
Spring Training Series
Georgia Tech

February 15, 2018

Today's Plan

1. Introduction to the STL
2. How to read cppreference.com
3. Common Containers
4. Common Algorithms
5. Code Examples

Introduction to the STL

Brief History

- 1979 - C++ Invented
- 1992 - STL Created
- 1998 - First Standardization

The STL

Standard **T**emplate **L**ibrary

- Containers
- Iterators
- Algorithms
- Function Objects

The STL

Algorithms $\rightarrow$ Iterators $\rightarrow$ Containers

Containers

- Store data (objects / primitives)
- The *Data Structures* of
  Data Structures and Algorithms (CS 1332)
- Minimal member methods for managing contents

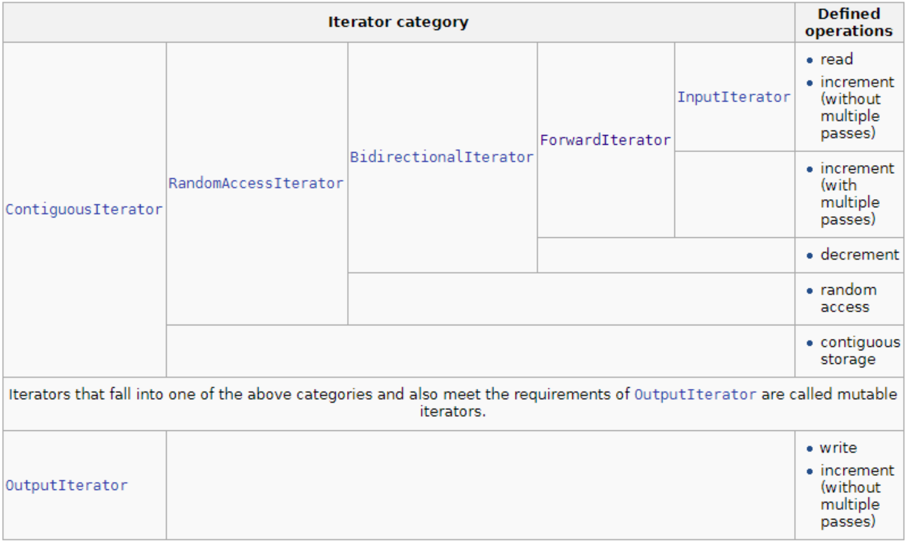
`http://en.cppreference.com/w/cpp/container`

Iterators

- Interface for useful container operations
- Exposed through begin() / end()
  (and their variants)

`http://en.cppreference.com/w/cpp/iterator`

Iterators



Iterator category					Defined operations
ContiguousIterator	RandomAccessIterator	BidirectionalIterator	ForwardIterator	InputIterator	• read • increment (without multiple passes)
					• increment (with multiple passes)
					• decrement
					• random access
					• contiguous storage
Iterators that fall into one of the above categories and also meet the requirements of OutputIterator are called mutable iterators.					
OutputIterator					• write • increment (without multiple passes)

Algorithms

- Utility functions for ranges of elements
- Decoupled from specific containers

`http://en.cppreference.com/w/cpp/algorithm`

Algorithms

Categories of Algorithms

- Non-Modifying
- Modifying
- Sorting / Partitioning
- Numeric

How to read cppreference.com

cppreference.com

`http://en.cppreference.com`

cppreference.com

http://en.cppreference.com

- Guided tour through std::vector's cppreference page.
- Constructors
- Member Functions
- Example

Common Containers

array

- Fixed size container
- Preferred over "c-style" arrays
- Two template arguments: *type* and *size*

Example

```
std::array<int,5> my_array = {0,1,2,3,4};
```

vector

set

map

Common Algorithms

any_of / all_of / none_of

copy

fill

generate

accumulate

transform

Code Examples

- TODO a handful of simple, but not trivial examples