

# STL

## C++'s Standard Containers Library

Matthew Barulic

RoboJackets  
Spring Training Series  
Georgia Tech

February 14, 2018

# Today's Plan

- 1 Introduction to the STL
- 2 How to read [cppreference.com](http://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

## Standard Template Library

- Containers
- Iterators
- Algorithms
- Function Objects

# The STL

Algorithms  $\rightarrow$  Iterators  $\rightarrow$  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`

- 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	<ul style="list-style-type: none"><li>• read</li><li>• increment (without multiple passes)</li></ul>
					<ul style="list-style-type: none"><li>• increment (with multiple passes)</li></ul>
					<ul style="list-style-type: none"><li>• decrement</li></ul>
					<ul style="list-style-type: none"><li>• random access</li></ul>
					<ul style="list-style-type: none"><li>• contiguous storage</li></ul>
Iterators that fall into one of the above categories and also meet the requirements of OutputIterator are called mutable iterators.					
OutputIterator					<ul style="list-style-type: none"><li>• write</li><li>• increment (without multiple passes)</li></ul>

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

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

## Categories of Algorithms

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

## How to read `cppreference.com`

`http://en.cppreference.com`

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

## Common Containers









# map

## Common Algorithms

`any_of` / `all_of` / `none_of`





generate



# accumulate

# transform

## Code Examples

- TODO a handful of simple, but not trivial examples