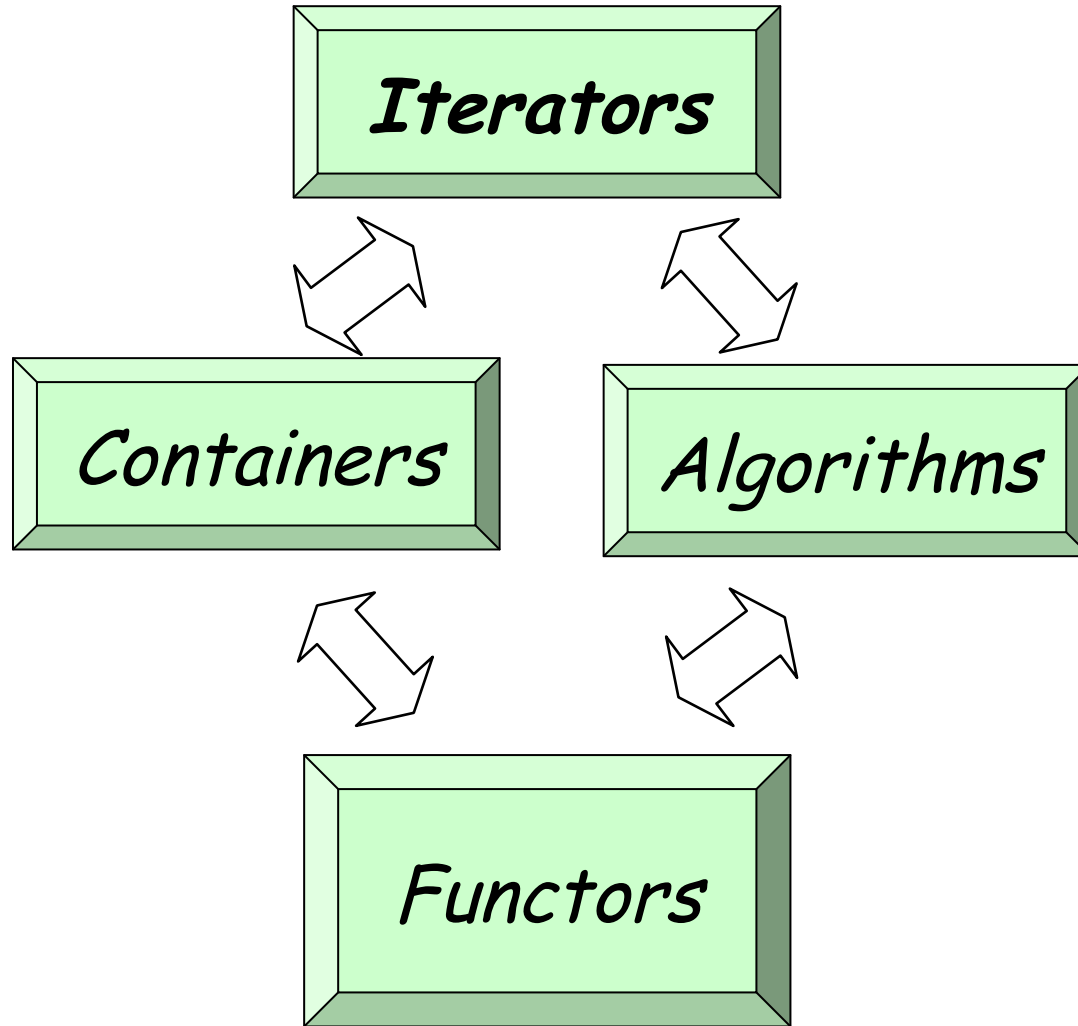


# **The Standard C++ Library – Iterators**

Version 1: Dr. Ofir Pele

Version 2: Dr. Erel Segal-Halevi

# Main Components



## *Why Iterators?*

- Instead of writing e.g. "find" for vector, "find" for unordered\_set, "find" for array, etc. -
- - we write only **one** find that accepts two iterators (begin and end):  
<http://www.cplusplus.com/reference/algorithm/find/>
- The same "find" would work for *any* container that defines the iterators correctly, and even for non-containers such as "range", "accumulate".

# *Iterator types*

**++,  
input**

**Input Iterator**

**++,  
output**

**Output Iterator**

**++, I/O**

**Forward Iterator**

**++, --, I/  
O**

**Bidirectional Iterator**

**Pointer  
arithmetic**

**Random-Access Iterator**

# Iterator Types

|            | Output              | Input                             | Forward                           | Bi-directional                    | Random   |
|------------|---------------------|-----------------------------------|-----------------------------------|-----------------------------------|--|
| Read       |                     | <code>x = *i</code>               | <code>x = *i</code>               | <code>x = *i</code>               | <code>x = *i</code>  |
| Write      | <code>*i = x</code> |                                   | <code>*i = x</code>               | <code>*i = x</code>               | <code>*i = x</code>  |
| Iteration  | <code>++</code>     | <code>++</code>                   | <code>++</code>                   | <code>++</code> , <code>--</code> | <code>++</code> , <code>--</code> , <code>+</code> , <code>-</code> ,<br><code>+=</code> , <code>-=</code>             |
| Comparison |                     | <code>==</code> , <code>!=</code> | <code>==</code> , <code>!=</code> | <code>==</code> , <code>!=</code> | <code>==</code> , <code>!=</code> , <code>&lt;</code> , <code>&gt;</code> ,<br><code>&lt;=</code> , <code>&gt;=</code> |

- Output: write only and can write only once
  - Input: read many times each item
  - Forward supports both read and write
  - Bi-directional support also decrement
  - Random supports random access
- (just like C pointer)

# Iterator types of containers

Input/output/forward iterators:

- iostreams (folder 1)

Bidirectional iterators:

- list, map, set

Random access iterators:

- vector

# IntBufferSwap example revisited

- See folder 2.
- Focus on iterator and `const_iterator`.