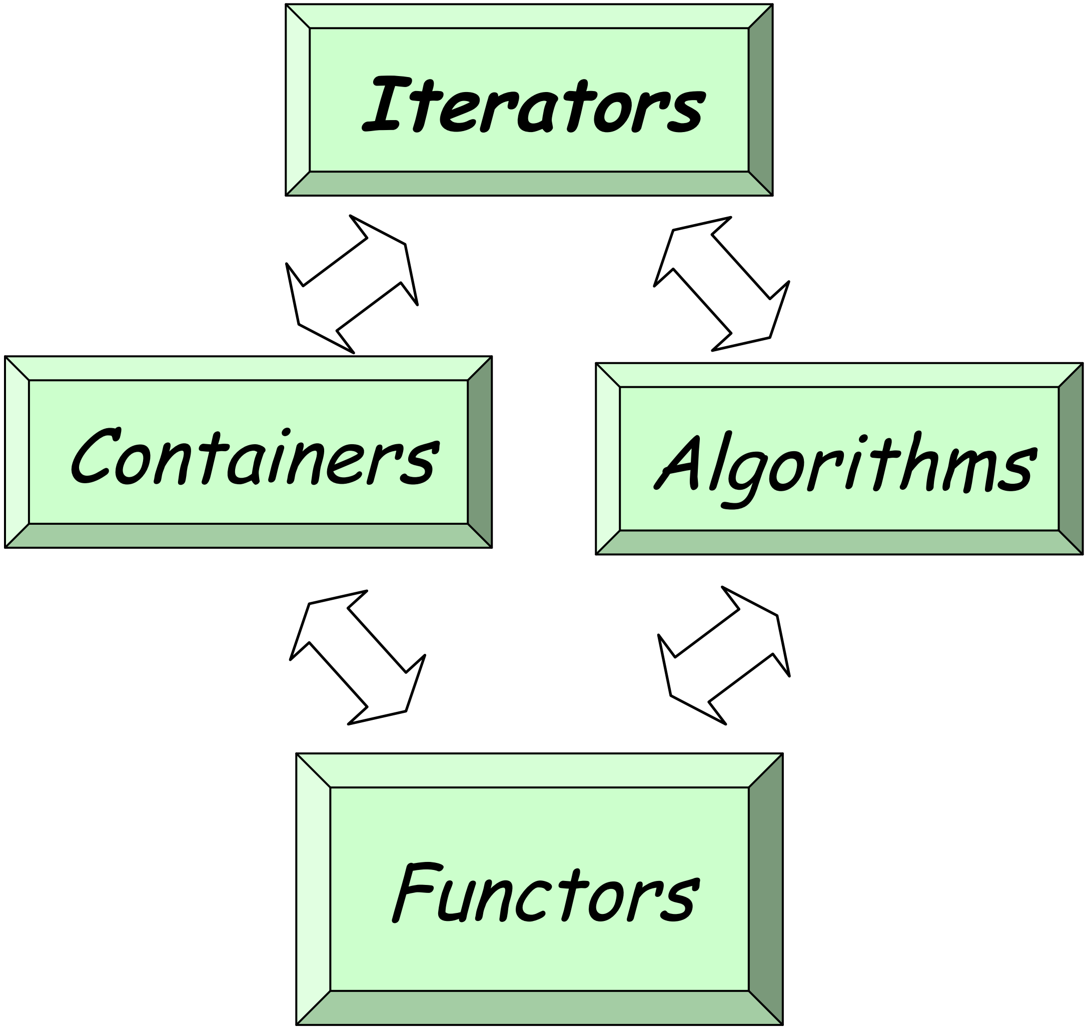
The Standard C++ Library – Iterators

Version 1: Dr. Ofir Pele
Version 2: Dr. Erel Segal-Halevi

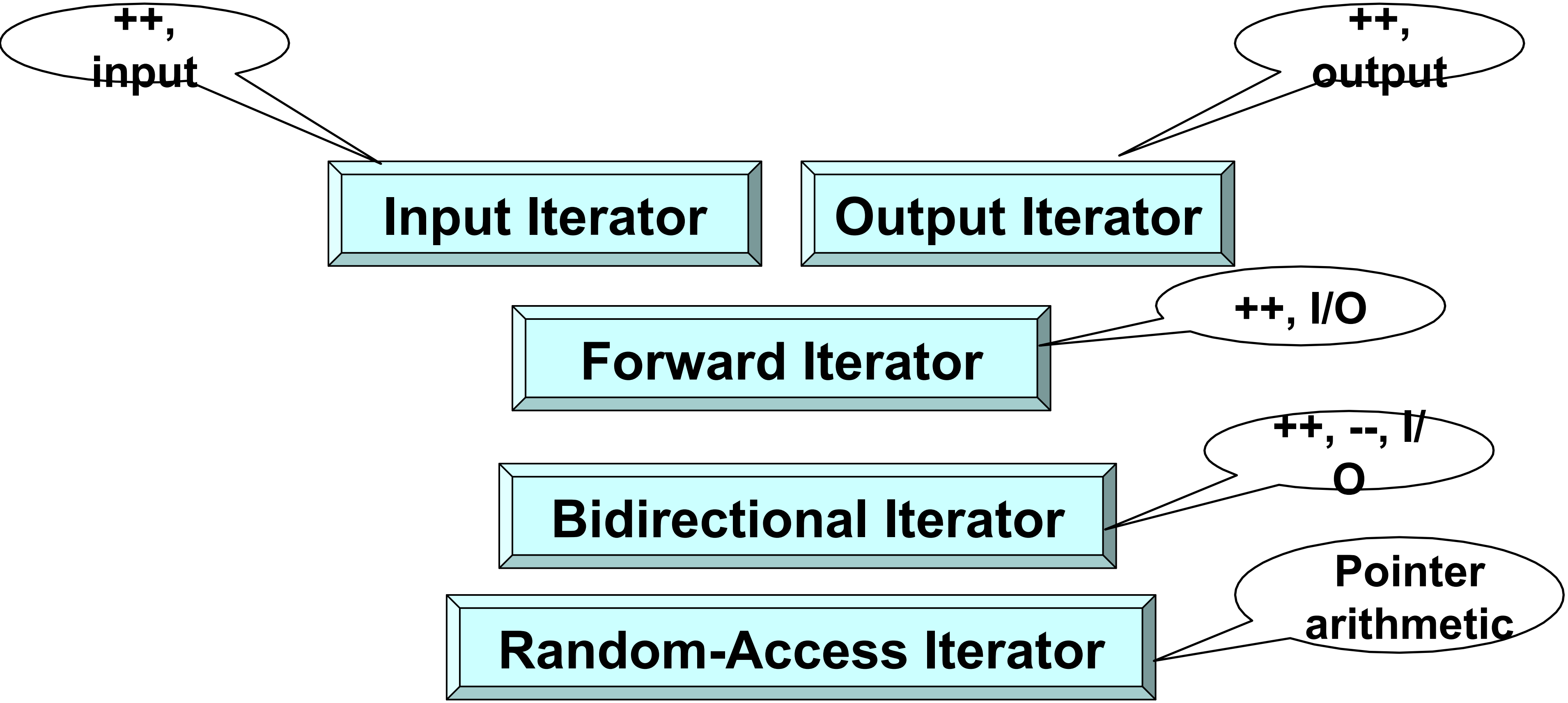
Main Components



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

Forward Iterator

++, I/O

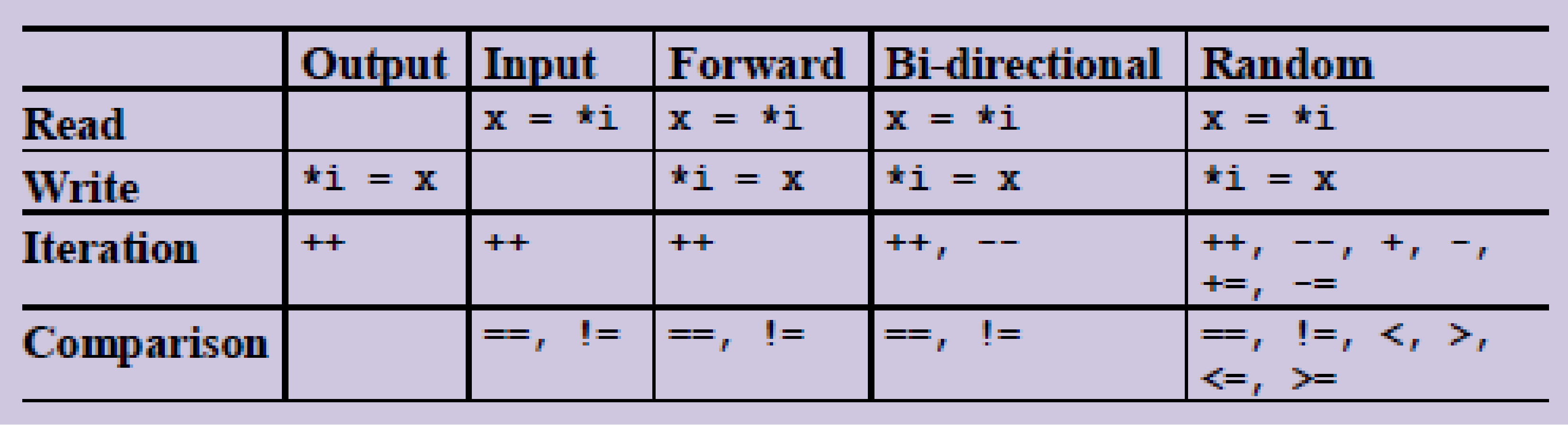
Bidirectional Iterator

++, --, I/O

Random-Access Iterator

Pointer arithmetic

Iterator Types



	Output	Input	Forward	Bi-directional	Random
Read		`x = *i`	`x = *i`	`x = *i`	`x = *i`
Write	`*i = x`		`*i = x`	`*i = x`	`*i = x`

Iteration	`++`	`++`	`++`	`++, --`	`++, --, +, -,`
					`+=, -=`

Comparison		`==, !=`	`==, !=`	`==, !=`	`==, !=, <, >,`
					`<=, >=`

- 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 iterators:

- iostreams (folder 1)

Forward iterator:

- forward_list

Bidirectional iterators:

- list, map, set

Random access iterators:

- vector, array

IntBufferSwap example revisited

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