STL BiDirectional Iterators

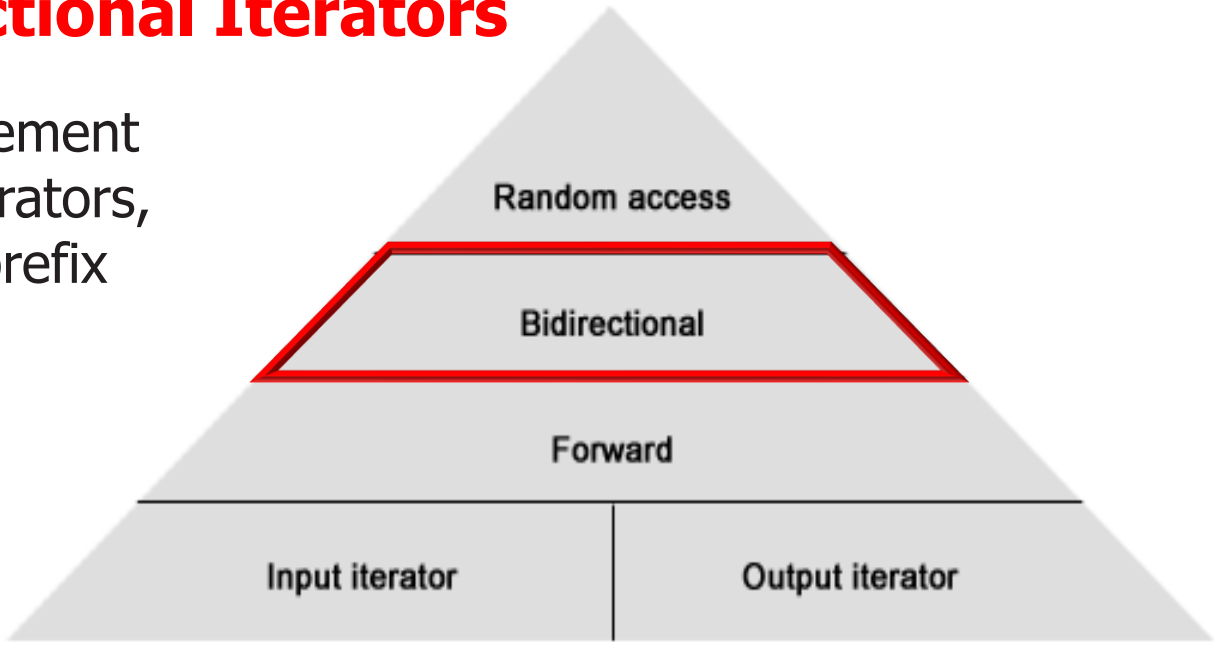
STL Bidirectional Iterators

- *Bidirectional* iterators allow algorithms to pass through the elements forward & backward

See www.cplusplus.com/reference/iterator/BidirectionalIterator

STL Bidirectional Iterators

- *Bidirectional* iterators must implement the requirements for *forward* iterators, plus the decrement operators (prefix & postfix)



STL Bidirectional Iterators

- Many STL containers implement *bidirectional* iterators

Associative Container	Keys sorted	Value available	Identical keys possible
`std::set`	Yes	No	No
`std::multiset`	Yes	No	Yes
`std::unordered_set`	No	No	No
`std::unordered_multiset`	No	No	Yes
`std::map`	Yes	Yes	No
`std::multimap`	Yes	Yes	Yes
`std::unordered_map`	No	Yes	No
`std::unordered_multimap`	No	Yes	Yes

STL list is also implements *bidirectional* iterators

STL Bidirectional Iterator Example

```
template <typename BidirectionalIterator, typename Compare>
void bubble_sort (BidirectionalIterator first,
                  BidirectionalIterator last, Compare comp) {
  BidirectionalIterator left_el = first, right_el = first;
  ++right_el;
  while (first != last) {
    while (right_el != last) {
      if (comp(*right_el, *left_el)) std::swap (left_el,
      right_el);
      ++right_el;
      ++left_el;
    }
    --last;
    left_el = first, right_el = first;
    ++right_el;
  }
}
```

See github.com/douglascraigschmidt/CPlusPlus/tree/master/STL/S-04/4.6