Swap

This topic teaches the swap algorithm by demonstrating the usage including swap, iter_swap and swap_ranges and explaining the algorithms

- Figure 16.7 demonstrates algorithms `swap`, `iter_swap` and `swap_ranges` for *swapping* elements.

`swap( a[ 0 ], a[ 1 ] );`

- Uses the `swap` algorithm to exchange two values
- The first and second elements of array `a` are exchanged
- The function takes as arguments references to the two values being exchanged

```
iter_swap( a.begin(), a.begin() + 1 );
```

- Uses `iter_swap` to exchange the two elements
- Takes two forward iterator arguments
  - In this case, iterators to elements of an array
  - Exchanges the values in the elements to which the iterators refer

```
swap_ranges(a.begin(), a.begin() + 5,
            a.begin() + 5 );
```

- Uses swap_ranges to exchange the elements from a.begin() up to, but *not* including, a.begin() + 5
- With the elements beginning at position a.begin() + 5
- The function requires three *forward iterator* arguments.

- The first two arguments specify the range of elements in the first sequence that will be exchanged with the elements in the second sequence starting from the iterator in the third argument
- Sequences can be from different arrays or containers
- Sequences must not overlap.
- The destination sequence must be large enough to contain all the elements of the ranges being swapped

This topic taught the swap algorithm by demonstrating the usage including swap, iter_swap and swap_ranges and explaining the algorithms