Reverse Iterators

- can convert with .base() member function

Insert Iterators

- NOTE: all are output iterators -> any algorithm that takes an output iterator can take one
- back inserter (vector, deque, list, string)
  - class: back_insert_iterator
  - called function: push_back
  - creation function: back_inserter(cont)

```
// standalone usage
std::back_insert_iterator<std::vector<int>> it{coll};
*it = val;

// or
auto it = std::back_inserter(coll);
*it = val;
```

- front inserter (available only for deque, list, and forward_list)
  - class: front_insert_iterator
  - called function: push_front
  - creation function: front_inserter(cont)

```
// standalone usage
std::front_insert_iterator<std::vector<int>> it{coll};
*it = val;

// or
auto it = std::front_inserter(coll);
*it = val;
```

- general inserter (not available for std::array or std::forward_list)
  - class: insert_iterator
  - called function: insert
  - creation function: inserter(cont)

```
// standalone usage
std::insert_iterator<std::vector<int>> it{coll, coll.begin() + n};
*it = val;

// or
```

```
auto it = std::inserter(coll, coll.begin() + n);
*it = val;
```

Stream Iterators

Move Iterators

- iterator that converts any access to underlying elements into move operations
- lets algorithms move from one container rather than copy

```
std::vector<std::string> dest{std::make_move_iterator(src.begin()), std::make_move
```