STL Forward
Iterators
STL Forward Iterators

- *Forward* iterators must implement the union of requirements for *input* & *output* iterators, plus a default ctor

See [www.cplusplus.com/reference/iterator/ForwardIterator](http://www.cplusplus.com/reference/iterator/ForwardIterator)
STL Forward Iterators

• A difference to output iterators is that operator* is also valid on the left side of operator=(*it = v is valid)
• Moreover, the # of assignments to a forward iterator is not restricted

```cpp
template <typename ForwardIterator, typename T>
void replace (ForwardIterator first, ForwardIterator last, const T& old_value, const T& new_value) {
    for (; first != last; ++first) {
        if (*first == old_value)
            *first = new_value;
    }
}
```
STL Forward Iterator Example

template <typename ForwardIterator, typename T>
void replace (ForwardIterator first, 
                ForwardIterator last, 
                const T& old_value, 
                const T& new_value) {
    for (; first != last; ++first)
        if (*first == old_value) *first = new_value;
}

// Initialize 3 ints to default value 1
std::vector<int> v (3, 1);
v.push_back (7); // vector v: 1 1 1 7
replace (v.begin(), v.end(), 7, 1);
assert (std::find (v.begin(), v.end(), 7) == v.end());