

STL Input Iterators

STL Input Iterators

- *Input* iterators are used to read values from a sequence

```
template<typename InputIterator,
        typename OutputIterator>
OutputIterator copy
    (InputIterator first,
     InputIterator last,
     OutputIterator result) {
    for (; first != last;
         ++first, ++result)
        *result = *first;
    return result;
}

vector<int> v;

copy (istream_iterator<int> (cin),
      istream_iterator<int>(),
      back_inserter(v));
```

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

STL Input Iterators

- They may be dereferenced to refer to some object & may be incremented to obtain the next iterator in a sequence

```
template<typename InputIterator,
        typename OutputIterator>
OutputIterator copy
    (InputIterator first,
     InputIterator last,
     OutputIterator result) {
    for (; first != last;
         ++first, ++result)
        *result = *first;
    return result;
}

vector<int> v;

copy (istream_iterator<int> (cin),
      istream_iterator<int>(),
      back_inserter(v));
```

STL Input Iterators

- An *input* iterator must allow the following operations
 - Copy ctor & assignment operator for that same iterator type
 - Operators == & != for comparison with iterators of that type
 - Operators * (can be const) & ++ (both prefix & postfix)

```
template<typename InputIterator,
        typename OutputIterator>
OutputIterator copy
    (InputIterator first,
     InputIterator last,
     OutputIterator result) {
    for (; first != last;
         ++first, ++result)
        *result = *first;
    return result;
}

vector<int> v;

copy (istream_iterator<int> (cin),
      istream_iterator<int>(),
      back_inserter(v));
```

STL Input Iterator Example

```
int main () {
    // An initially empty vector.
    vector<int> v;

    // copy contents of cin as "int" and store at the end of vector v.
    for (istream_iterator<int> i (cin);
         i != istream_iterator<int> ();
         ++i)
        // Add int to the end of the vector.
        v.push_back (*i);

    // Use STL copy() algorithm along with back_inserter()!
    copy (istream_iterator<int> (cin),
          istream_iterator<int>(),
          back_inserter(v));
}
```

See github.com/douglascraigsschmidt/CPlusPlus/tree/master/STL/S-04/4.3/4.3a