

- Use a range library (Boost or ASL)

```
auto p = find(begin(a), end(a), x);  
auto p = find(a, x);
```

- Have many variants of simple, common algorithms such as find() and copy()

- Look for interface symmetry

```
sort(a, [](const employee& x, const employee& y){ return x.last < y.last; });  
auto p = lower_bound(a, "Parent", [](const employee& x, const string& y){ return x.last < y; });  
  
sort(a, less(), &employee::last);  
auto p = lower_bound(a, "Parent", less(), &employee::last);
```