Assignment #1, Part b)

- Questions re. sample solution?

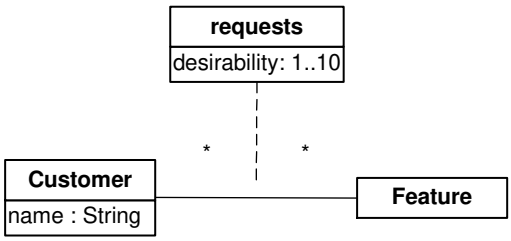
Q. Which OOA should I use as a basis for my OOD?

- sample solution
- your original
- a modified version of yours (but then hand it in)

Q. How should I start?

- Take the OOA
  - decide how the use case will get done
    - sequence diagram
  - decide which associations and classes to keep, and what implementation space classes and associations to add (e.g., Input.read(filename))
  - decide on navigability of associations
  - implement classes, attributes, associations, using
    - Java classes, methods, data members, static data members
  - implement operations as methods
  - done!

Example



- From requirements
  - Chance of a report listing all features requested by a customer is high.
  - with each feature, print "how badly customers want the feature"
    - if few customers, will give their names

Navigability

- From Feature will need to traverse efficiently to all request objects.
- From request, will need to traverse efficiently to the customer.
  - N.B. can traverse other way, but if we don't store pointers to help us do it, it will be very inefficient!
- May want to change in the future, so keep interface general.

Association Class

```
public class CustomerRequestsFeature {
    public CustomerRequestsFeature(Customer c, Feature f, int d){
        desirability = d;
        customer = c;
        f.addCustomerRequest(this);
    }

    public Customer getCustomer() { return customer; }
    public int getDesirability() { return desirability; }

    private int desirability;
    private Customer customer;
}
```

- Do we need to provide APIs + implementations to get the customer?
  - YES
- Do we need to provide APIs + implementations to get the feature?
  - NO, not for the uses cases we are implementing.

Customer Class

```
public class Customer {
    public Customer(String n){
        name = n;
    }

    public String getName() { return name; }

    private String name;
}
```

- Do we need APIs + implementation to get all the requests associated with this customer?
  - NO
- Do we need access to the name?
  - YES
- Do we need to change the name after creation?
  - NO

Feature Class

```
public class Feature {
    …
    /*package*/ void addCustomerRequest(CustomerRequestsFeature cr){
        customerRequests.add(cr)
    }
    private Vector customerRequests = new Vector();
}
```

- Do we need implementation to get all requests?
  - YES
- Do we need a method to get all the customer requests?
  - NO
    - Operations on sequence diagram shows all traversal of the relationship will be done internally in the Feature class.