# **Establishing Relationships**

Deborah Kurata <a href="http://msmvps.com/blogs/deborahk/">http://msmvps.com/blogs/deborahk/</a>
<a href="mailto:@DeborahKurata">@DeborahKurata</a>
<a href="mailto:deborahk@insteptech.com">deborahk@insteptech.com</a>





## **Object-Oriented Programming (OOP)**

# Identifying Classes



- Represents business entities
- Defines properties (data)
- Defines methods (actions/behavior)

# Separating Responsibilities



- Maximizes cohesion
- Minimizes coupling
- Simplifies Maintenance
- Improves Testability

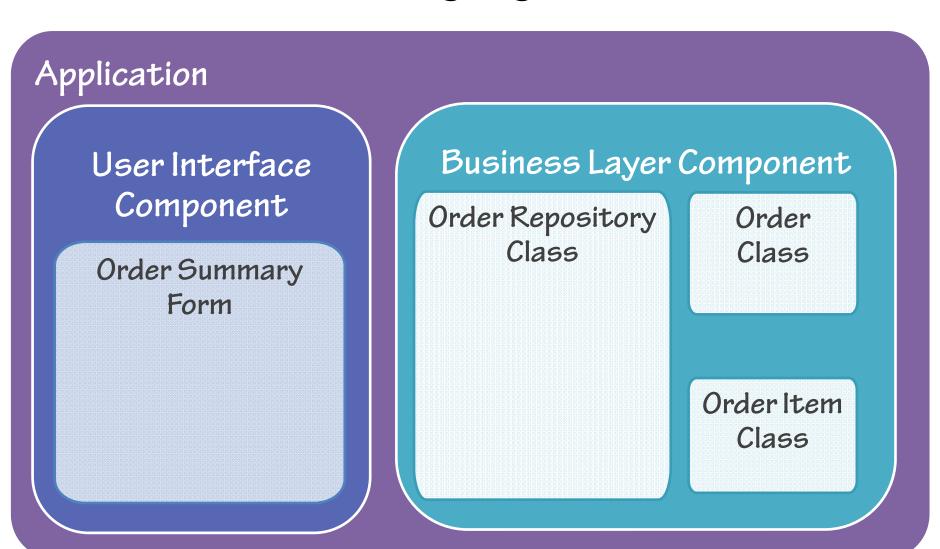
Establishing Relationships



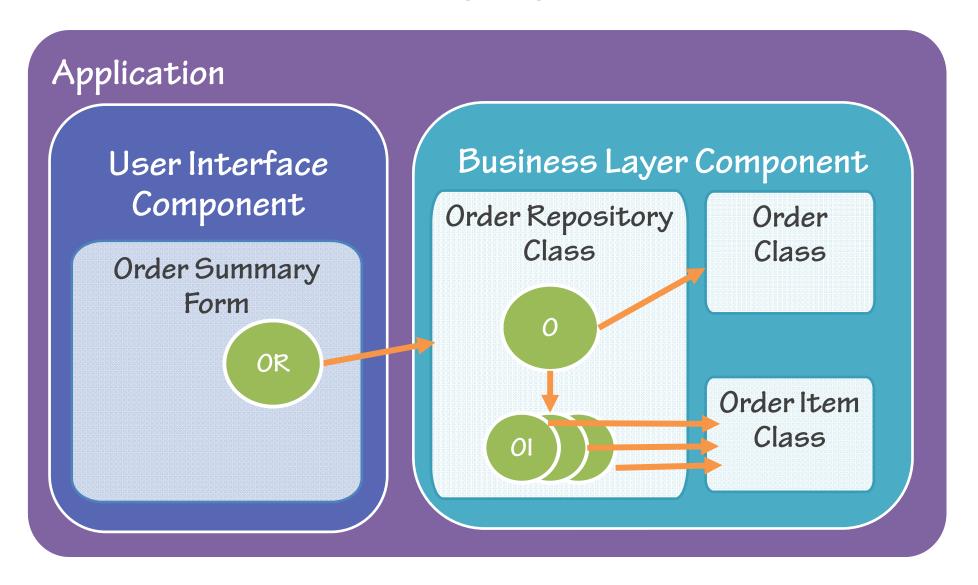
 Defines how objects work together to perform the operations of the application

Leveraging Reuse

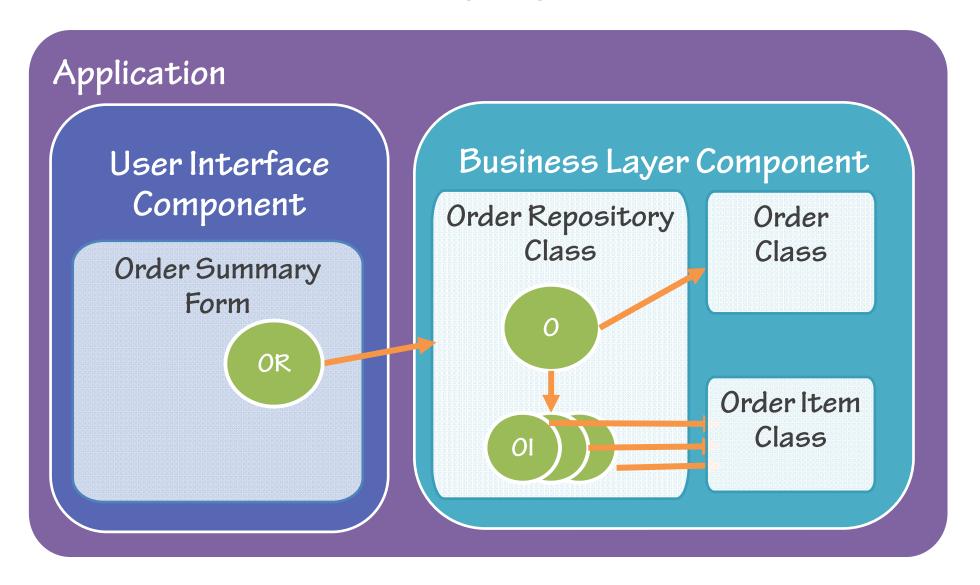
## **Working Together**



# **Working Together**



# **Working Together**



#### **Module Outline**

Defining the Relationships

Types of Relationships

Collaboration

Composition

Composition: References

Composition: Ids

**Inheritance** 

## **Defining the Relationships**

#### Customer

- Name
- Email address
- Home address
- Work address
- Validate()

#### **Product**

- Product name
- Description
- Current price
- Validate()

### Order

- Customer
- Order date
- Shipping address
- Validate()

#### Order Item

- Product
- Quantity
- Purchase price
- Validate()
- Retrieve()
- Save()

# **Customer Repository**

- Retrieve()
- Save()

# Product Repository

- Retrieve()
- Save()

# Order Repository

- Retrieve()
- Save()

#### **Address**

- Street Line 1
- Street Line 2
- City
- State/Prov.
- Postal Code
- Country ...

## **Defining the Relationships**

Order Repository

Order

Address Repository

**Address** 

Order Item

**Customer Repository** 

Customer

Product Repository

**Product** 

## **Object Relationships**

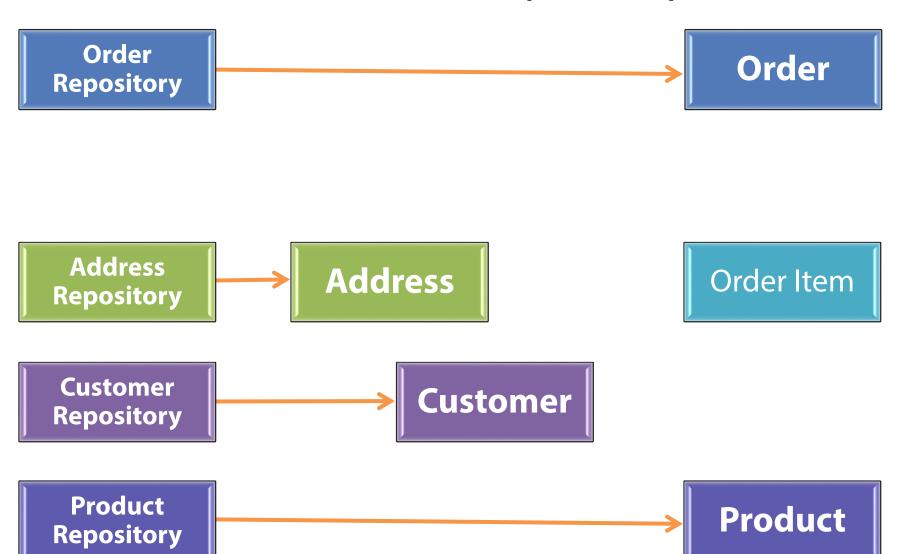
Collaboration ("uses a")

Composition ("has a")

Inheritance ("is a")

Aggregation

### **Collaboration ("uses a")**



## Composition ("has a")

Order Repository

Address Repository

**Customer Repository** 

Product Repository

Address

### **Customer**

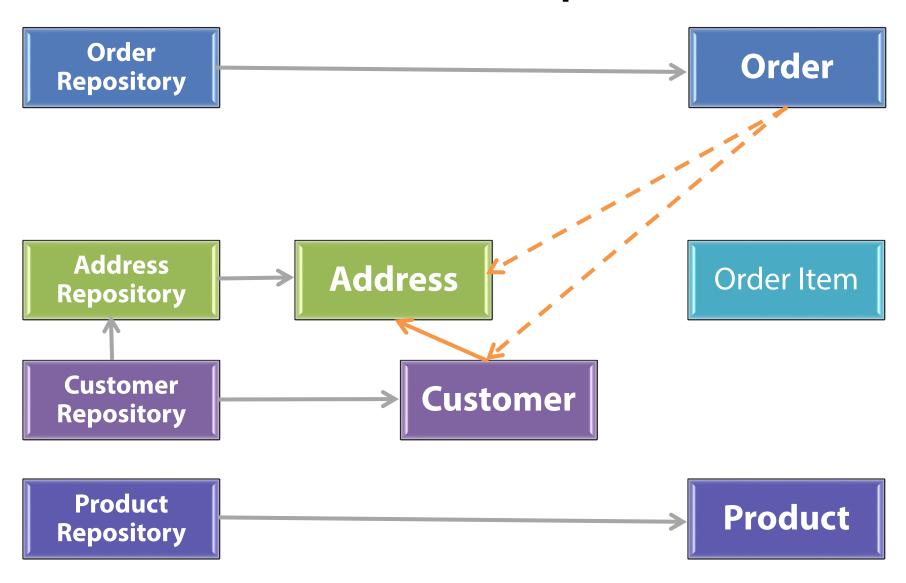
- Name
- Email address
- Home address
- Work address
- Validate()

#### Order

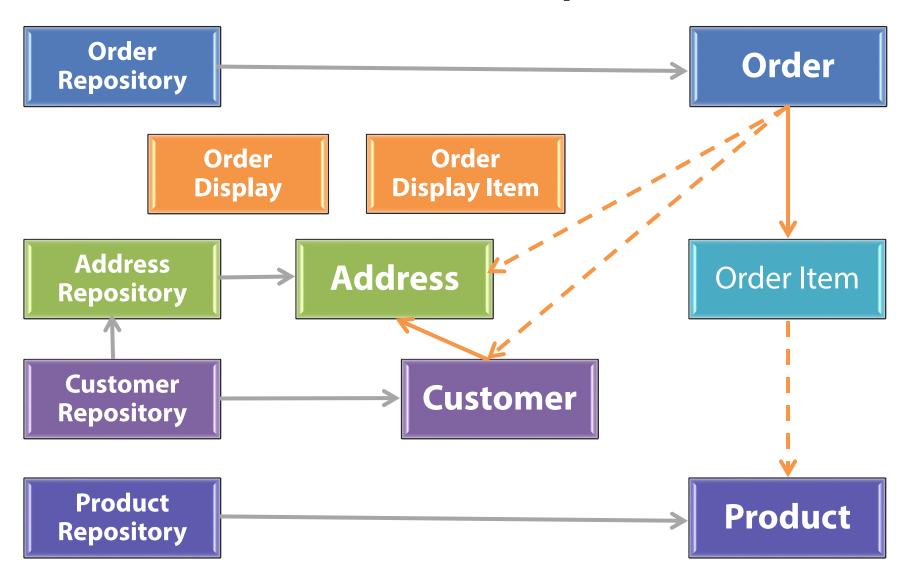
- Customer
- Order date
- Shipping address
- Validate()
- Product
- Quantity
- Purchase price
- Validate()
- Retrieve()
- Save()

**Product** 

## Relationships



## Relationships

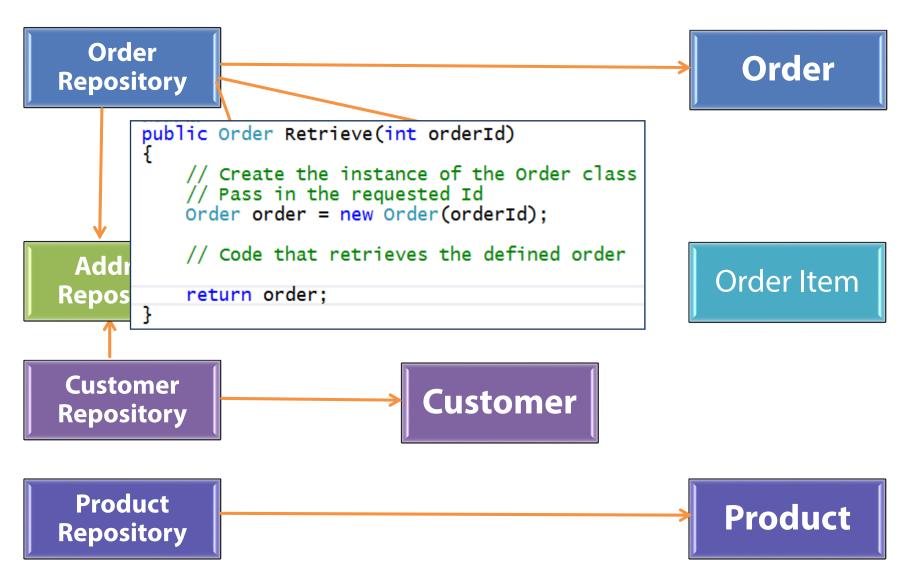


## From the Requirements

"The new system must manage business, residential, government, and educator types of customers."

"The new system must manage business, **Inheritance** residential, government, and educator types of customers." Customer Business Residential Educator Government Local Elementary **High School** State **Federal** University

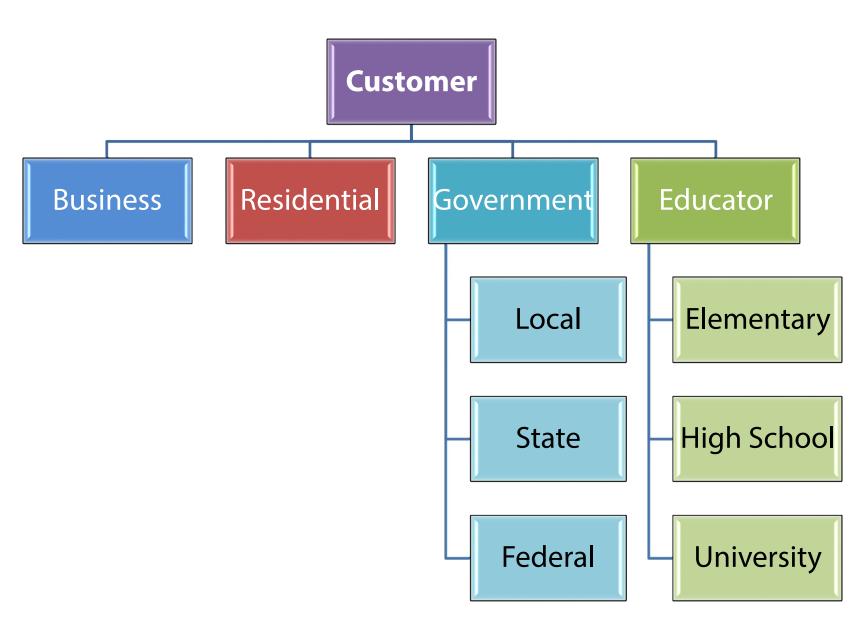
### **Collaboration ("uses a")**



### Composition ("has a")

```
public int CustomerId { get; set; }
                                                              Order
public DateTimeOffset? OrderDate { get; set; }
public int OrderId { get; private set; }
public List<OrderItem> orderItems { get; set; }
public int ShippingAddressId { get; set; }
  Address
                         Address
                                                            Order Item
 Repository
  Customer
                                  Customer
 Repository
               public Customer(int customerId)
   Product
                   this.CustomerId = customerId;
                                                             Product
                   AddressList = new List<Address>();
 Repository
               15 references | 1/1 passing
               public List<Address> AddressList { get; set; }
```

## **Inheritance** ("is a")



## **Four Pillars of OOP**

