

LECTURE 6

DATA MODELING

LEK HSIANG HUI

LEARNING OBJECTIVES

At the end of this lecture you should understand:

- What structural modeling is
- How to perform structural modeling using domain model class diagram

INTRODUCTION TO STRUCTURAL MODELING

Introduction
to Structural
Modeling

Introduction
to Entity
Class

Class
Diagram
Notation

Domain
Model Class
Diagram

STRUCTURAL MODELING



What is the
system?

During the
analysis phase,
we also do
Structural
Modeling

What are the
components?

An abstract
representation of
the system

STRUCTURAL MODEL

Main diagram = **class diagram**

Will be performing object-oriented structural analysis:

- Attributes (of business objects)
- Operations (of business objects)
- Numerical relationships (between business objects)
 - E.g. How many customers may co-own a particular account

Focuses on the “**nouns**” of the system

STRUCTURAL MODEL

Ensures internal consistency within the BRD

Each use case description is verified against the structural model

STEPS IN STRUCTURAL ANALYSIS

1. Identify **Entity Classes** and Add Attributes
2. Model Generalizations
3. Model Whole/Part Relationship
4. Analyze Associations
5. Analyze Multiplicity

INTRODUCTION TO ENTITY CLASS

Introduction
to Structural
Modeling

Introduction
to Entity
Class

Class
Diagram
Notation

Domain
Model Class
Diagram

STEPS IN STRUCTURAL ANALYSIS



1. Identify **Entity Classes**
and add Attributes

WHAT TO KEEP TRACK?



If you are building an e-commerce website, what do you need to keep track/record?

List of transactions?

List of Members?

ENTITY CLASSES

Entity Classes
= Business
objects that
need to be
recorded

Transaction,
Member are
examples of
Entity Classes

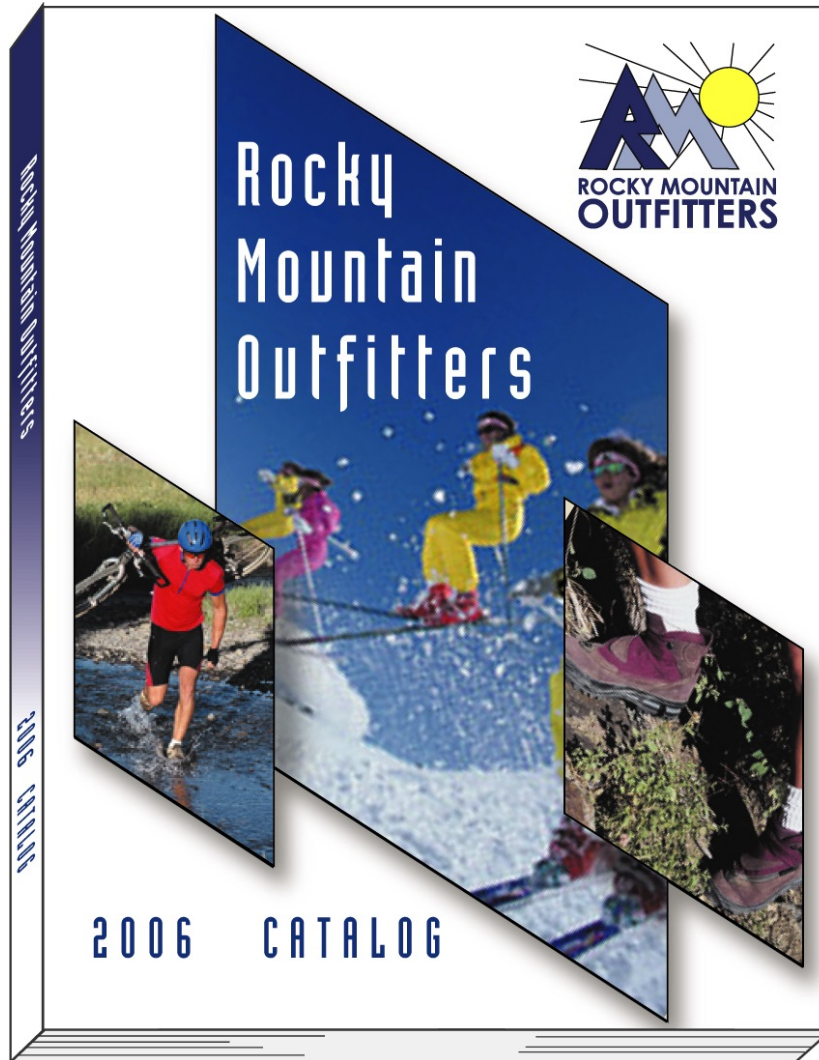
Need to
understand
the problem
domain

PROCEDURE FOR DEVELOPING AN INITIAL LIST OF THINGS

Review the **use case documentation** and **human-interface requirements**

- Any **noun phrase** appearing in these are potential entities
- E.g. Member, Transaction, Item, Person, Address, etc

ROCKY MOUNTAIN OUTFITTER CASE STUDY



Rocky Mountain Outfitters (RMO) is a sports clothing distributor

- Significant regional distributor
- Employs more than 600 people
- \$100 million annual sales

PARTIAL LIST OF “THINGS” BASED ON NOUNS FOR RMO

IDENTIFIED NOUN	NOTES ON INCLUDING NOUN AS A THING TO STORE
Accounting	We know who they are. No need to store it.
Back order	A special type of order? Or a value of order status? Research.
Back-order information	An output that can be produced from other information.
Bank	Only one of them. No need to store.
Catalog	Yes, need to recall them, for different seasons and years. Include.
Catalog activity reports	An output that can be produced from other information. Not stored.
Catalog details	Same as catalog? Or the same as product items in the catalog? Research.
Change request	An input resulting in remembering changes to an order.
Charge adjustment	An input resulting in a transaction.

PARTIAL LIST OF “THINGS” BASED ON NOUNS FOR RMO

IDENTIFIED NOUN	NOTES ON INCLUDING NOUN AS A THING TO STORE
Color	One piece of information about a product item.
Confirmation	An output produced from other information. Not stored.
Credit card information	Part of an order? Or part of customer information? Research.
Customer	Yes, a key thing with lots of details required. Include.
Customer account	Possibly required if an RMO payment plan is included. Research.
Fulfillment reports	An output produced from information about shipments. Not stored.
Inventory quantity	One piece of information about a product item. Research.
Product item	Yes, what RMO includes in a catalog and sells. Include.
Management	We know who they are. No need to store.
Marketing	We know who they are. No need to store.

PARTIAL LIST OF “THINGS” BASED ON NOUNS FOR RMO

IDENTIFIED NOUN	NOTES ON INCLUDING NOUN AS A THING TO STORE
Merchandising	We know who they are. No need to store.
Order	Yes, a key system responsibility. Include.
Payment method	Part of an order. Research.
Price	Part of a product item. Research.
Promotional materials	An output? Or documents stored outside the scope? Research.
Prospective customer	Possibly same as customer. Research.
Return	Yes, the opposite of an order. Include.
Return confirmation	An output produced from information about a return. Not stored.
RMO	There is only one of these! No need to store.
Season	Part of a catalog? Or is there more to it? Research.
Shipment	Yes, a key thing to track. Include. <i>(continued)</i>

ATTRIBUTES OF THINGS

Specific details of things are called **attributes**

Identifier (key): attribute uniquely identifying thing

- Examples: vehicle ID number, or product ID number

ALL CUSTOMERS HAVE THESE ATTRIBUTES:	EACH CUSTOMER HAS A VALUE FOR EACH ATTRIBUTE:		
Customer ID	101	102	103
First name	John	Mary	Bill
Last name	Smith	Jones	Casper
Home phone	555-9182	423-1298	874-1297
Work phone	555-3425	423-3419	874-8546

CLASS DIAGRAM NOTATION

Introduction
to Structural
Modeling

Introduction
to Entity
Class

Class
Diagram
Notation

Domain
Model Class
Diagram

NAMING CONVENTIONS

Person Table

id	name	age	contact_num	...
1	John	20	91234567	...
2	Mary	18	97654321	...
...				

Attribute list

Person

id
name
age
contactNum
...

getId()
getAge()
...

Class name

Name class with a
singular noun
(e.g. Student
instead of
Students)

For all the names,
make sure you follow
how you would do it in
the sourcecodes
(e.g. no spaces)

Method list

INTERVIEW QUESTIONS FOR FINDING CLASSES

What types of people and organizations do the system keep track of?

- Example: Customer, CardHolder, and BoardMember

What events and transactions does the system keep a record of?

- Example: Promotion, Sale

What products and services does the system keep a record of?

- Example: Products (keep the stock count for accounting)

STEPS IN STRUCTURAL ANALYSIS



2. Model Generalizations

RECALL: GENERALIZATION AND SPECIALIZATION

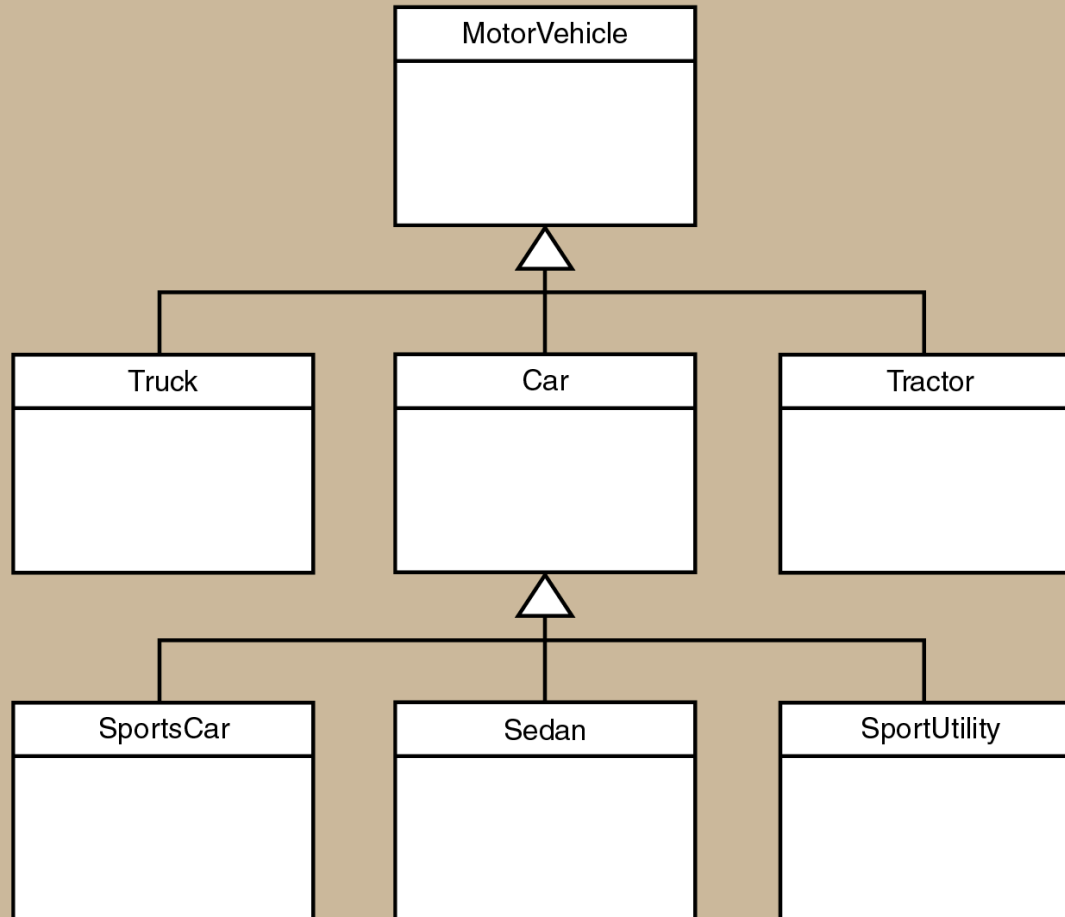
Specialized object inherit the ability to do all the things that a generalized object can do

Think Object-Oriented programming

Clerk extends (inherits) Employee

- Employee = Generalized actor (Superclass)
- Clerk = Specialized actor (Subclass)

A GENERALIZATION/SPECIALIZATION HIERARCHY FOR MOTOR VEHICLES



Trucks, Cars, and Tractors are special types of Motor Vehicles

Sports Cars, Sedans, and Sport Utilities are special types of Cars

STEPS IN STRUCTURAL ANALYSIS



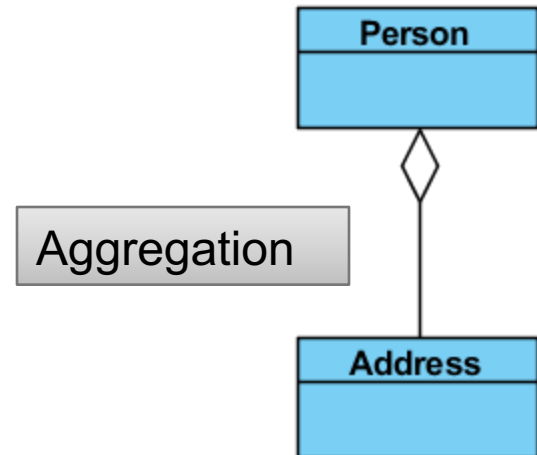
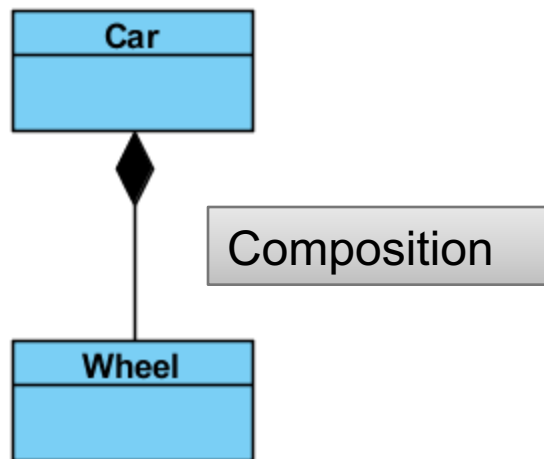
3. Model whole/part relationship

MODEL WHOLE/PART RELATIONSHIPS

Some objects consist of other objects

Model these relationships using **composition** and **aggregation**

- Composition and Aggregation describe the relationship between a whole and its parts

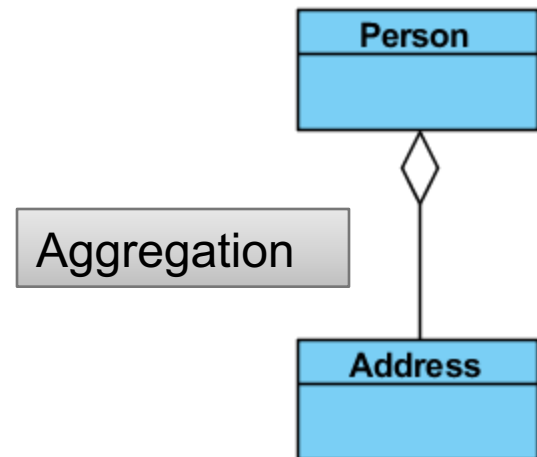
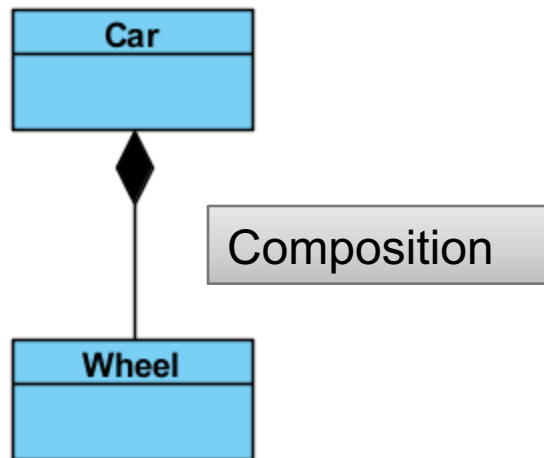


MODEL WHOLE/PART RELATIONSHIPS

Composition means that **the whole owns the part entirely**

- the part may not belong simultaneously to any other whole

Aggregation means that **there is some kind of whole/part relationship**

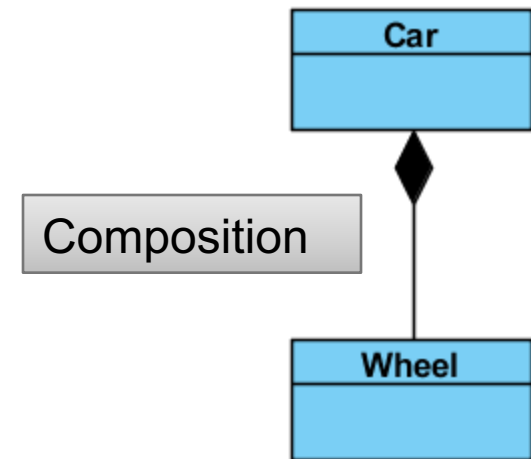


COMPOSITION

Car has one (or more) Wheel(s)

When a Car object is created, the associated Wheel objects are created (together with the Car)

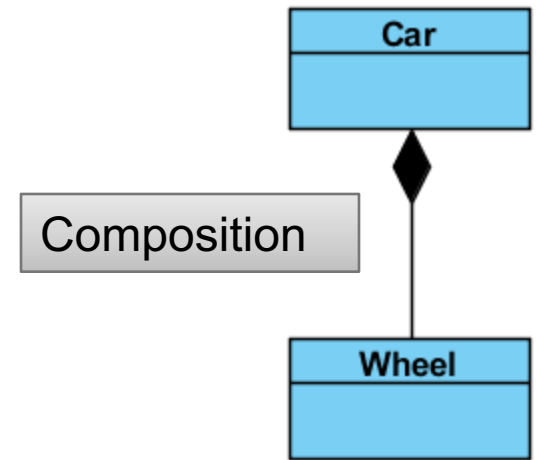
When a Car object is destroyed, the associated Wheel objects are also destroyed



COMPOSITION

```
class Wheel:
    def __init__(self, num):
        self.num = num

    def __str__(self):
        return repr("Wheel " + str(self.num))
```



COMPOSITION

```
from wheel import Wheel
```

```
class Car:
```

```
    wheels = []
```

```
    def __init__(self):
```

```
        self.wheels.append(Wheel(1))
```

```
        self.wheels.append(Wheel(2))
```

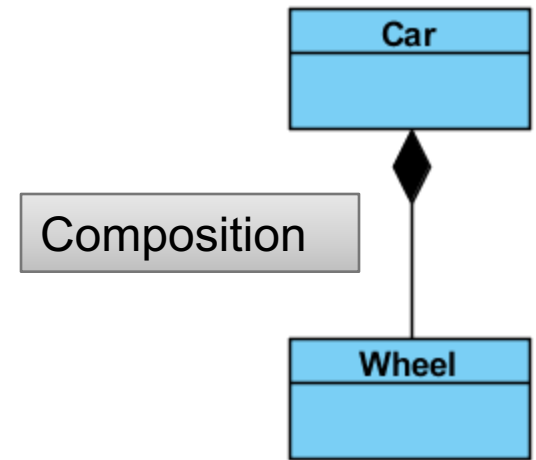
```
        self.wheels.append(Wheel(3))
```

```
        self.wheels.append(Wheel(4))
```

```
    def __str__(self):
```

```
        return repr("Car:: Wheels : " +
```

```
                    ', '.join(str(w) for w in self.wheels))
```

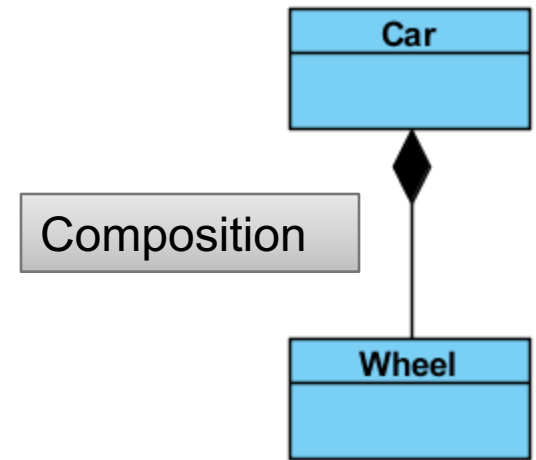


COMPOSITION

```
from car import Car
```

```
#wheels created with the Car  
c = Car()
```

```
print(c)
```

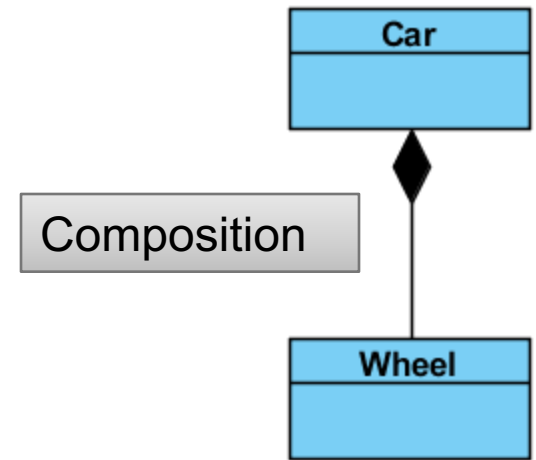


"Car:: Wheels : 'Wheel 1', 'Wheel 2', 'Wheel 3', 'Wheel 4'"

COMPOSITION (JAVA)

```
public class Wheel{
    . . .
}

public class Car{
    ArrayList wheels;
    public Car(){
        wheels = new ArrayList();
        wheels.add(new Wheel());
        wheels.add(new Wheel());
        wheels.add(new Wheel());
        wheels.add(new Wheel());
    }
}
```

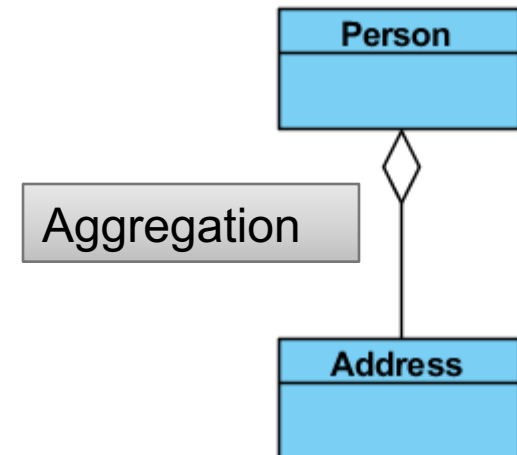


AGGREGATION

Person has one (or more) Address(es)

When a Person object is destroyed/ceased to exist, the associated Address object is **not** destroyed

Think of it as John (Person) bought a house (Address), this house is added to the John “object”. John died (assume the John object is destroyed), the house (Address) is not automatically destroyed because he died



AGGREGATION

```
class Address:
```

```
    def __init__(self, street, unit, postal):
```

```
        self.street = street
```

```
        self.unit = unit
```

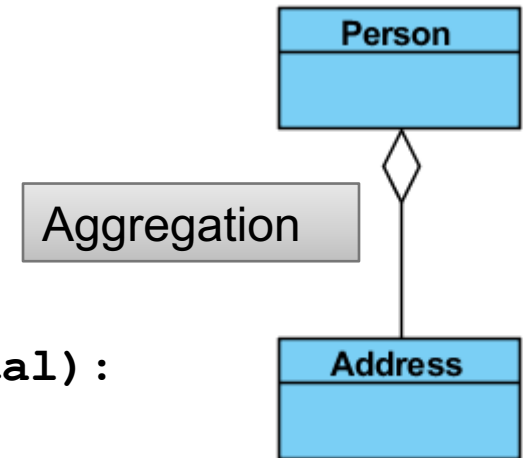
```
        self.postal = postal
```

```
    def __str__(self):
```

```
        return repr("Street : " + self.street
```

```
                    + ", Unit : " + self.unit
```

```
                    + ", Postal : " + self.postal)
```

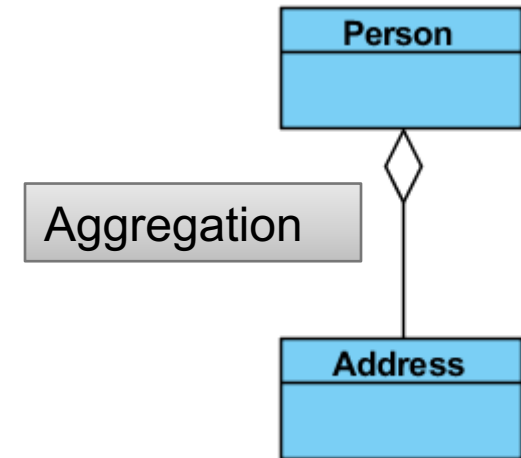


AGGREGATION

```
class Person:
```

```
    def __init__(self, address):  
        self.address = address
```

```
    def __str__(self):  
        return repr("Person with address : "  
                    + str(self.address))
```



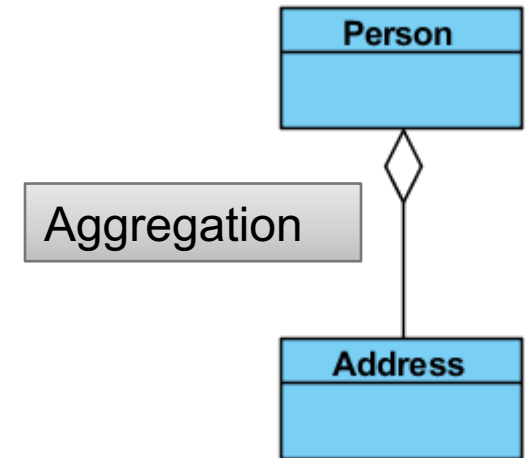
AGGREGATION

```
from person import Person
from address import Address
```

```
add1 = Address("13 Computing Drive", "#01-01", "117417")
```

```
john = Person(add1)
```

```
print(john)
```

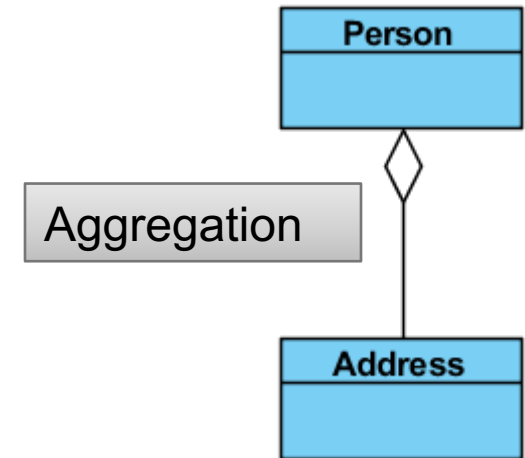


"Person with address : 'Street : 13 Computing Drive, Unit :
#01-01, Postal : 117417"

AGGREGATION (JAVA)

```
public class Address{  
    . . .  
}
```

```
public class Person{  
    private Address address;  
    public Person (Address address) {  
        this.address = address;  
    }  
}
```



STEPS IN STRUCTURAL ANALYSIS



4. Analyze associations

STEPS IN STRUCTURAL ANALYSIS



Discover the ways how one business object links with others

- Each of these relationships is an association

ANALYZE ASSOCIATIONS

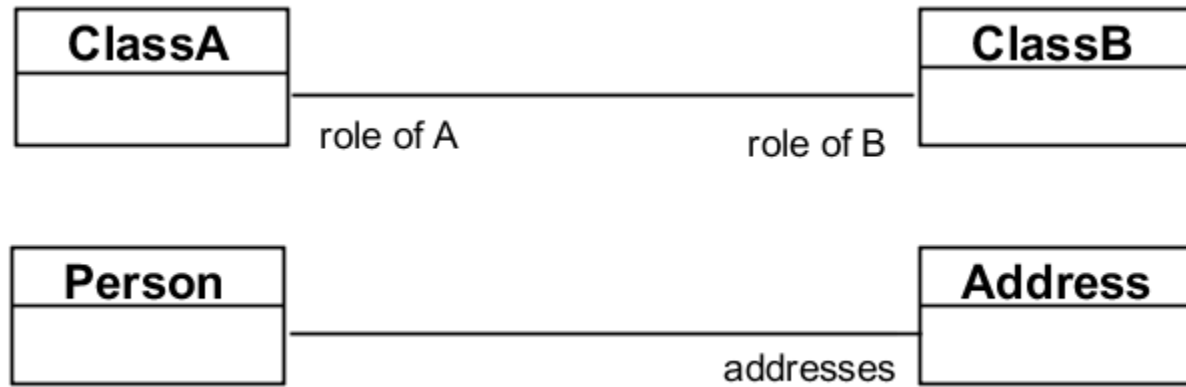
Information about one object refers to information about another object

- e.g. invoice data refers to product information (such as description and price), so Invoice is associated with Product

To carry out a business operation relating to one object, the operation affects related objects

- e.g. when a booking is canceled (an operation of Booking), flights must be updated to reflect the newly available seat. Booking is thus associated with Flight

INDICATE ASSOCIATIONS



```
class Address:  
    ...
```

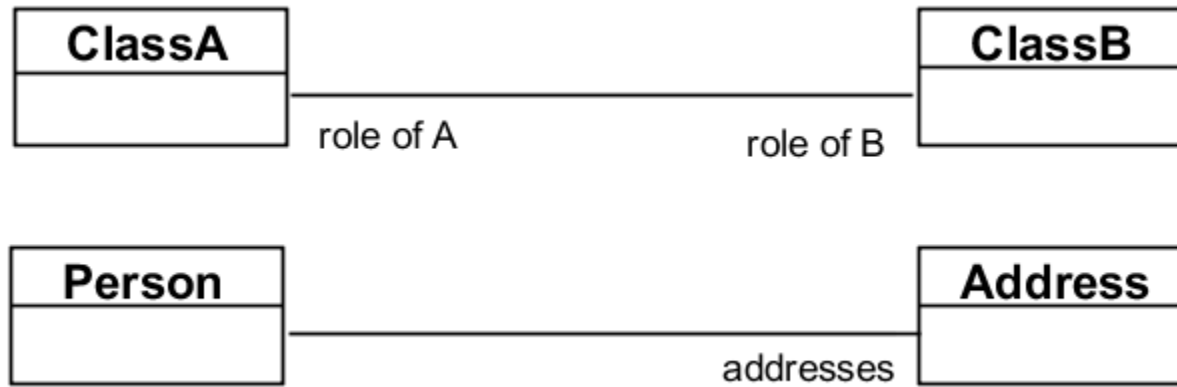
rolename

```
class Person:  
    def __init__(self, addresses):  
        self.addresses = addresses  
    ...
```

Or

```
class Person:  
    addresses = []  
    ...
```


INDICATE ASSOCIATIONS (JAVA)

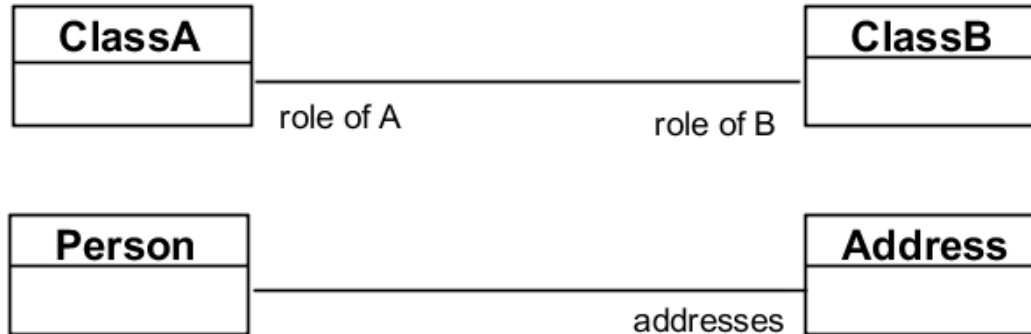


```
public class Address{  
    . . .  
}
```

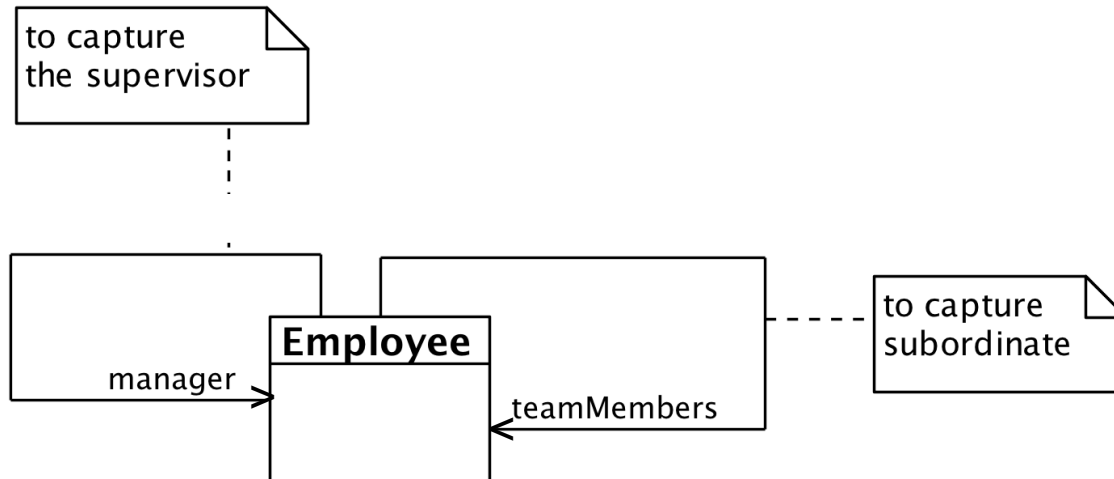
```
public class Person{  
    private Address[] addresses;  
    ...  
}
```

rolename

INDICATE ASSOCIATIONS



Binary
Association



Reflexive
Association

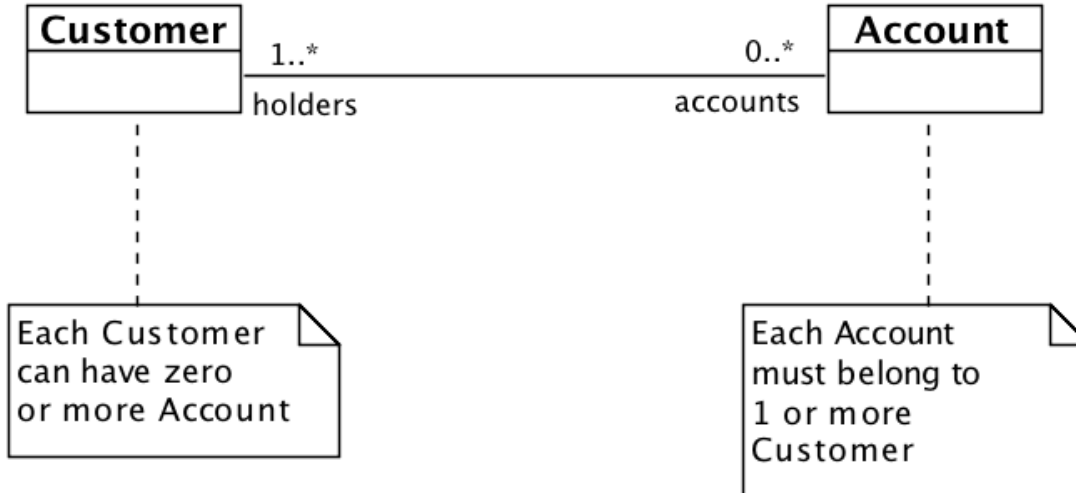
STEPS IN STRUCTURAL ANALYSIS



5. Analyze Multiplicity

- Model business rules that deal with the number of business objects that may be linked to each other

INDICATE MULTIPLICITY



1	Exactly one
*	Unlimited number (zero or more)
0..*	Zero or more
1..*	One or more
0..1	Zero or one
3..5	Specific Range (3 to 5 inclusive)

DOMAIN MODEL CLASS DIAGRAM

Introduction
to Structural
Modeling

Introduction
to Entity
Class

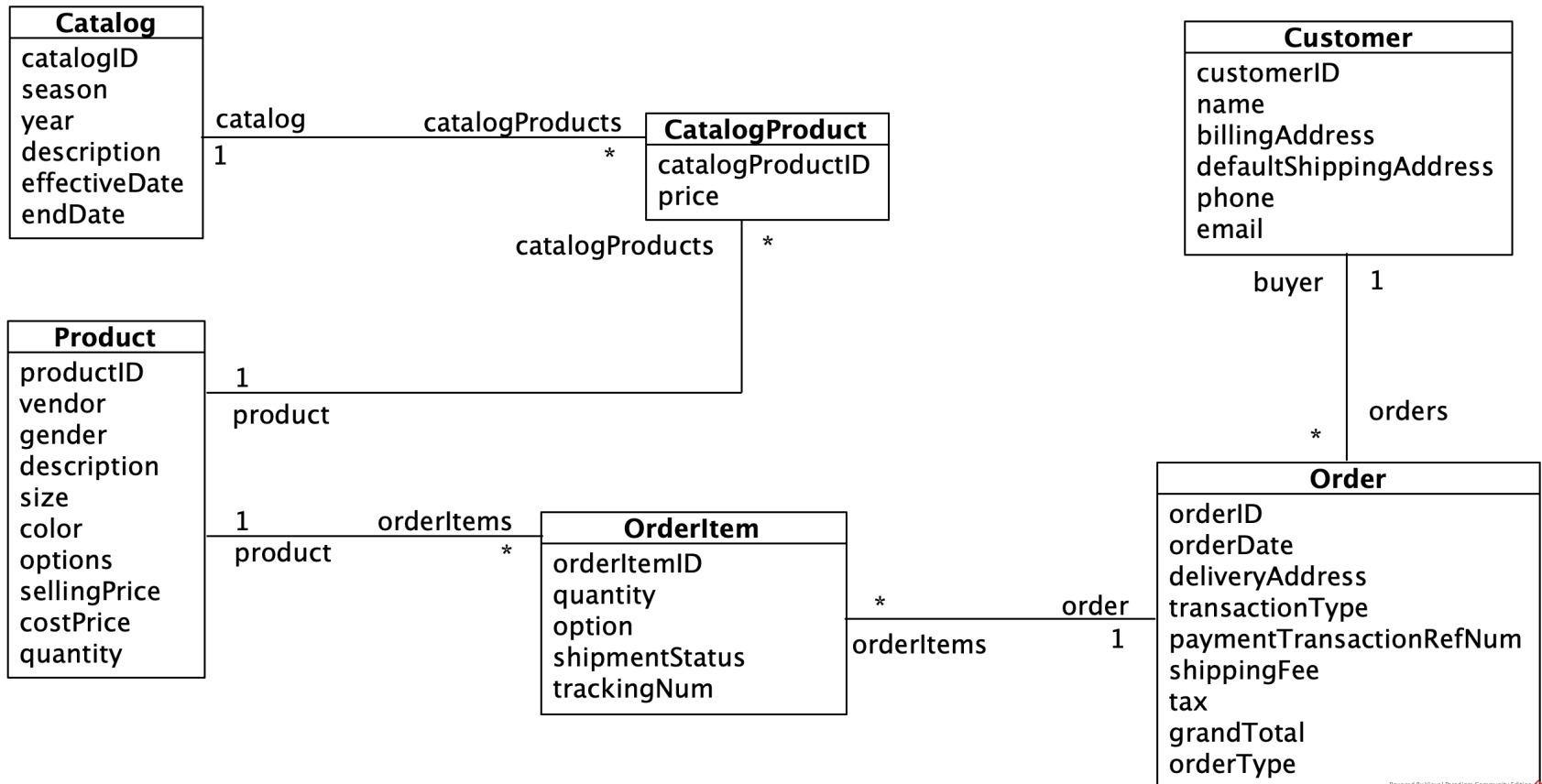
Class
Diagram
Notation


Domain
Model Class
Diagram

SIMPLIFIED DOMAIN MODEL CLASS DIAGRAM OF RMO

Domain Model Class Diagram only contains the Classes, Attributes, Rolenames, Associations

It doesn't have the operations/methods



A man with dark hair and glasses is looking directly at the camera with a slight smile. He is holding a white rectangular card in front of his face with both hands. The card has the text "It's your turn..." written on it in a black serif font. His fingers are visible at the top and bottom edges of the card. The background is a plain, light-colored wall.

It's your turn...

TASK

IS2102 Exam Question (Nov/Dec 2012 Q45)

SUMMARY

Structural Modeling using UML Class Diagram (Domain Model Class Diagram)

WHAT'S NEXT?

Modeling States