



Domain Models

Part 2

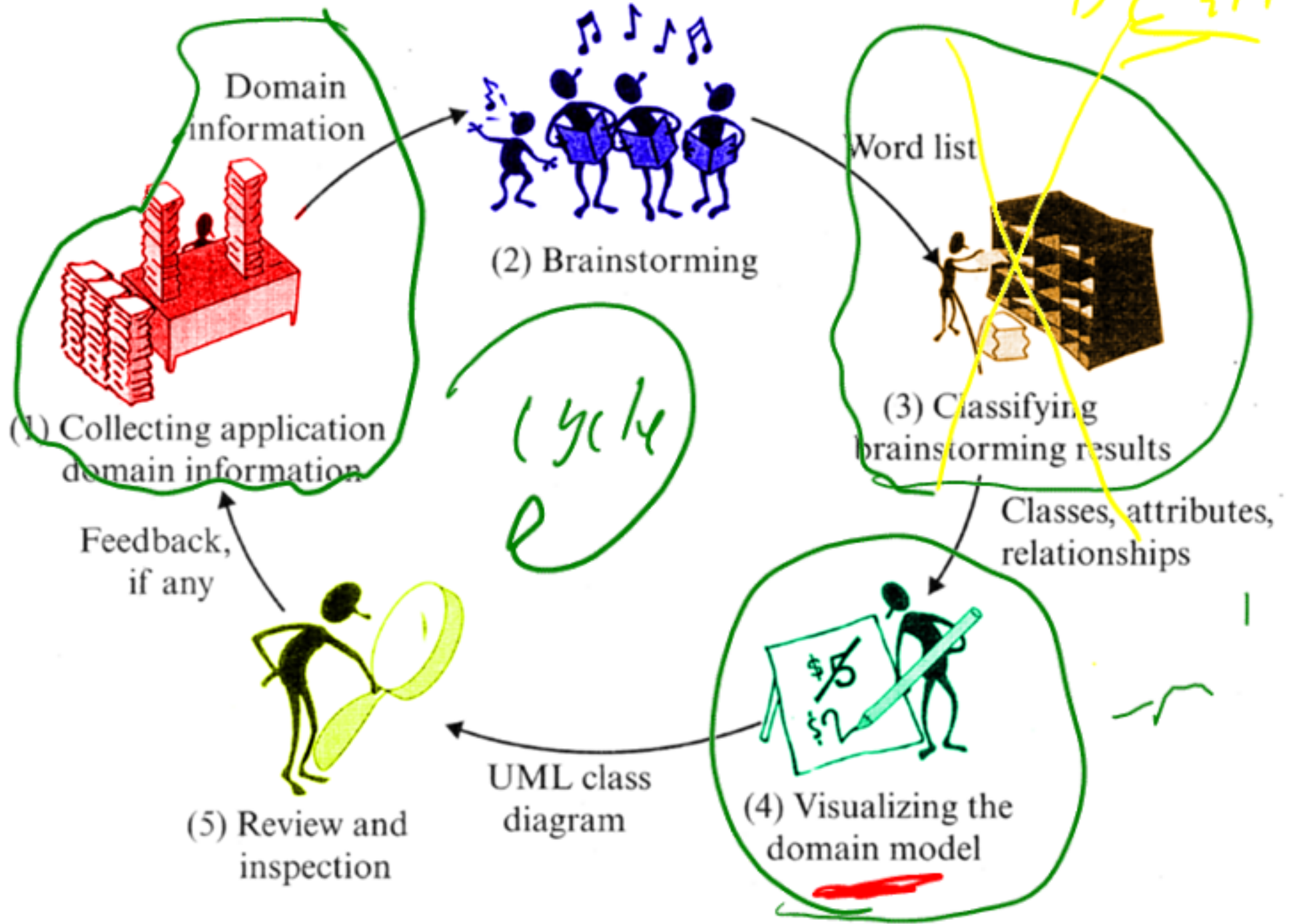
SE 1021

Objectives

- Discuss Data Structures & Implementation (Lab Review)
- Define aggregation and inheritance in terms of UML.
- Critique a model for correct usage of UML.
- Explain the difference between compact and expanded versions of UML.
- Review Aggregation and composition in UML (Lab Review)
- Construct an object domain model for a given problem based upon a use case

Domain Modeling

(Kung: Object oriented Software Engineering)



Definitions

- Aggregation
 - An aggregation is a binary relationship between two classes. It states that objects of one class are parts of objects of the other class.
- Inheritance
 - Inheritance is a binary relationship between two concepts or classes such that one concept or class is a generalization (or specialization) of the other.

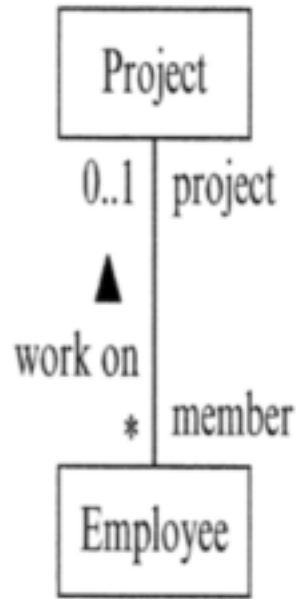
two

pieces of the other system
objects

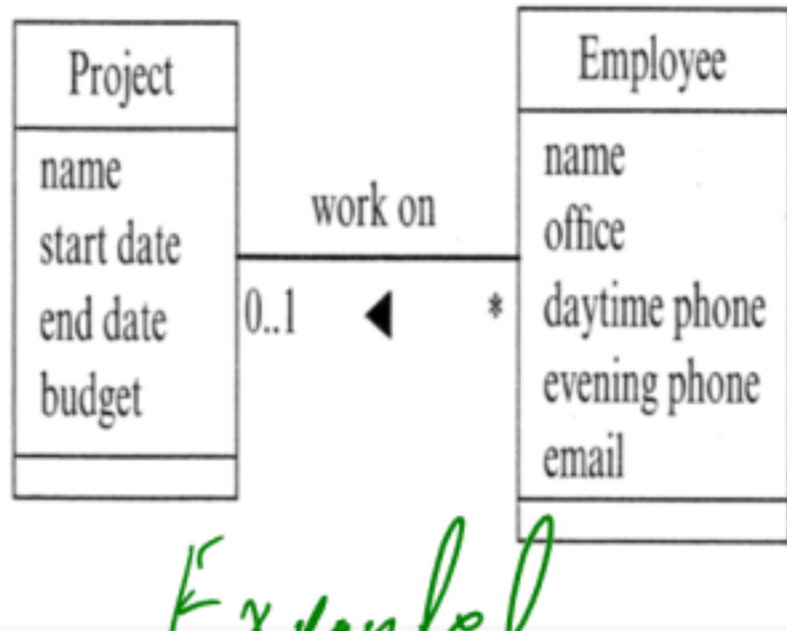
class gains behavior
attributes from another class



Compact versus expanded UML views for class diagrams



Compact
⇒ Used for higher level analysis. Focuses on relations between classes.

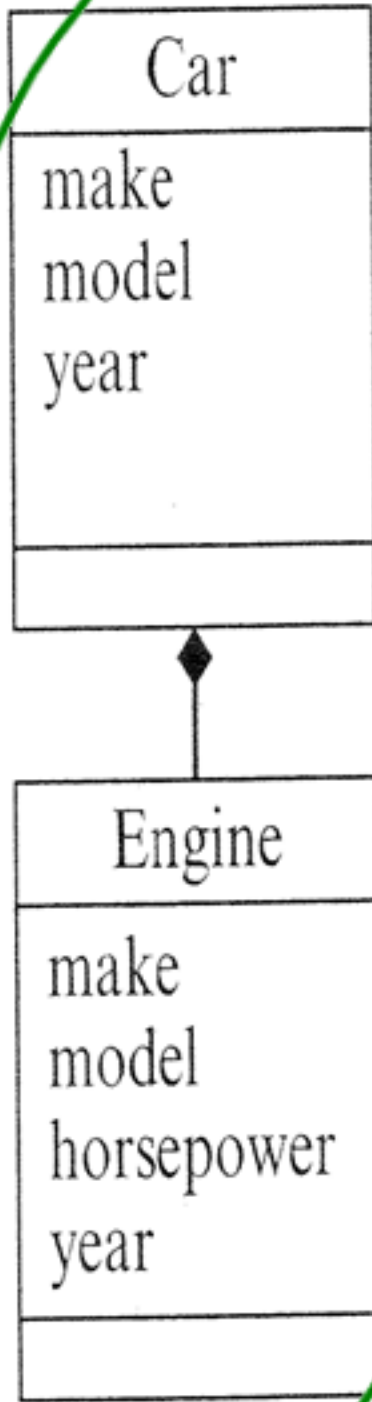


Expanded
⇒ More detailing in Expanded version.

Tends to be used later on in project.

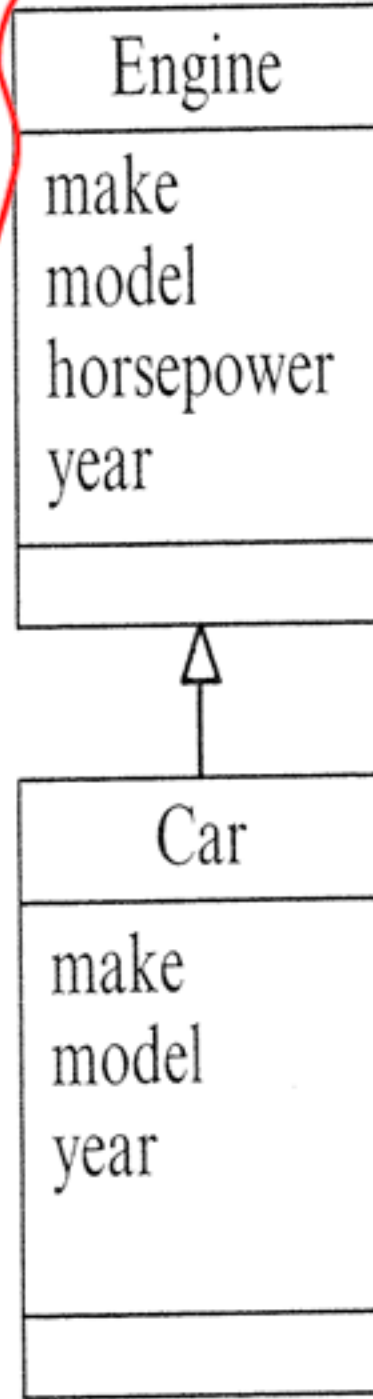


Which model is right?



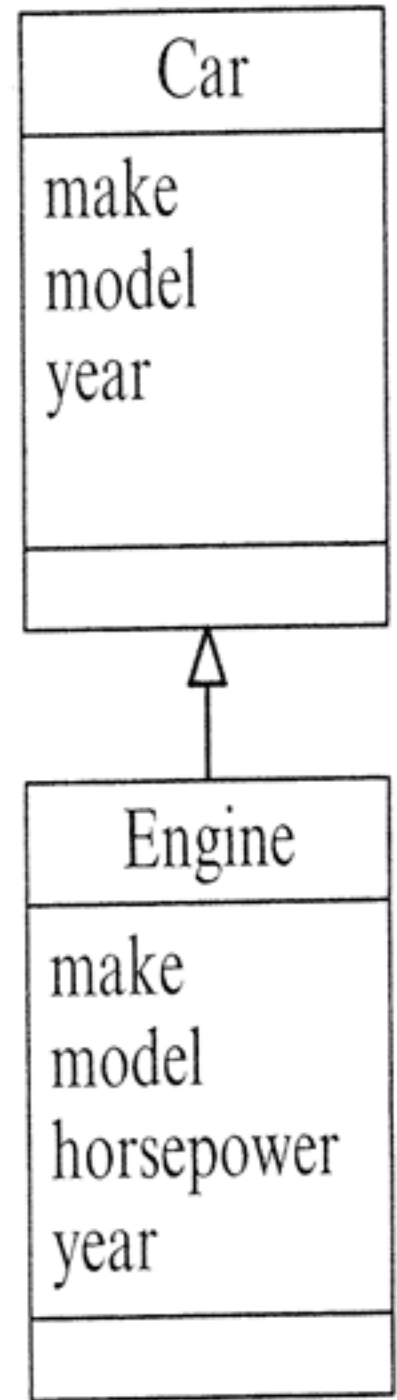
A

4 correct



B

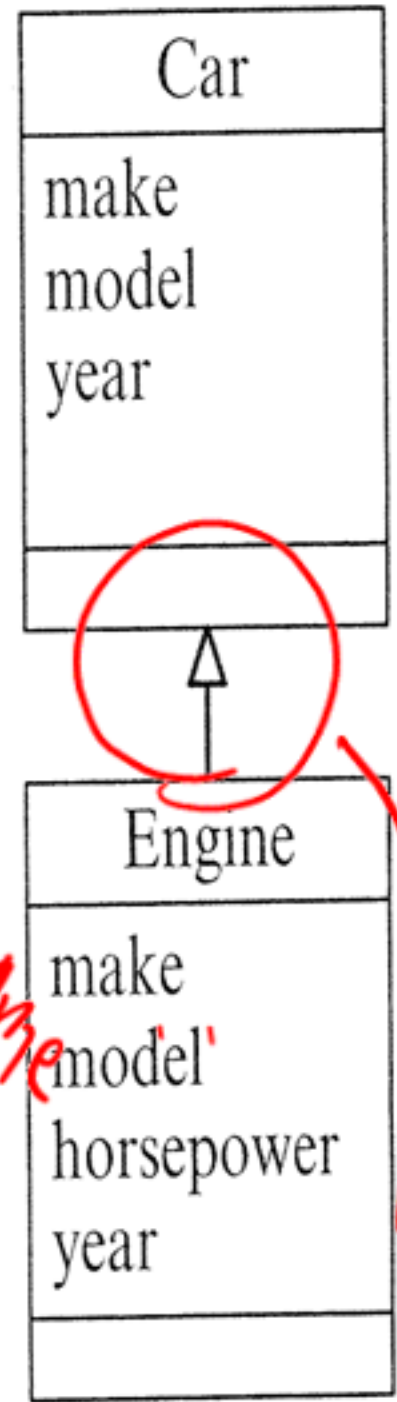
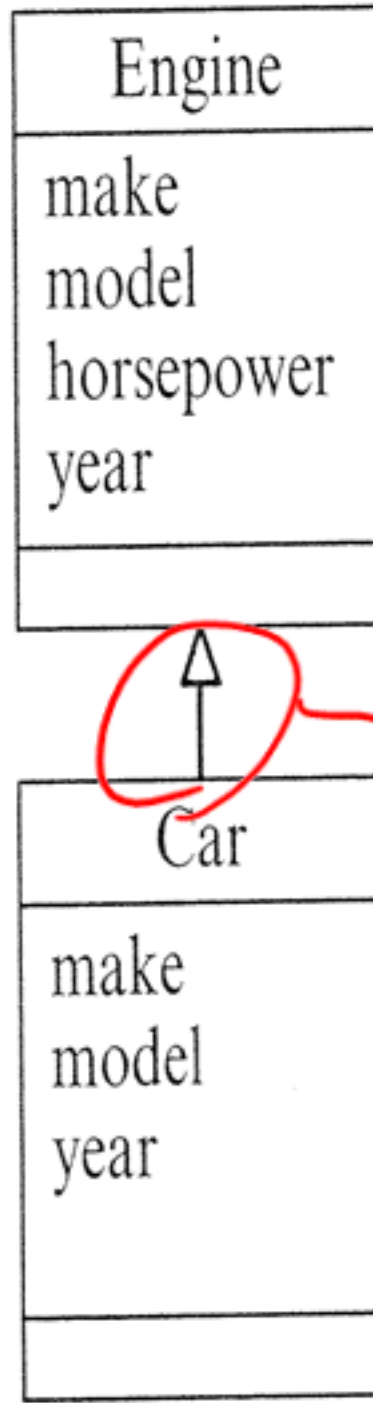
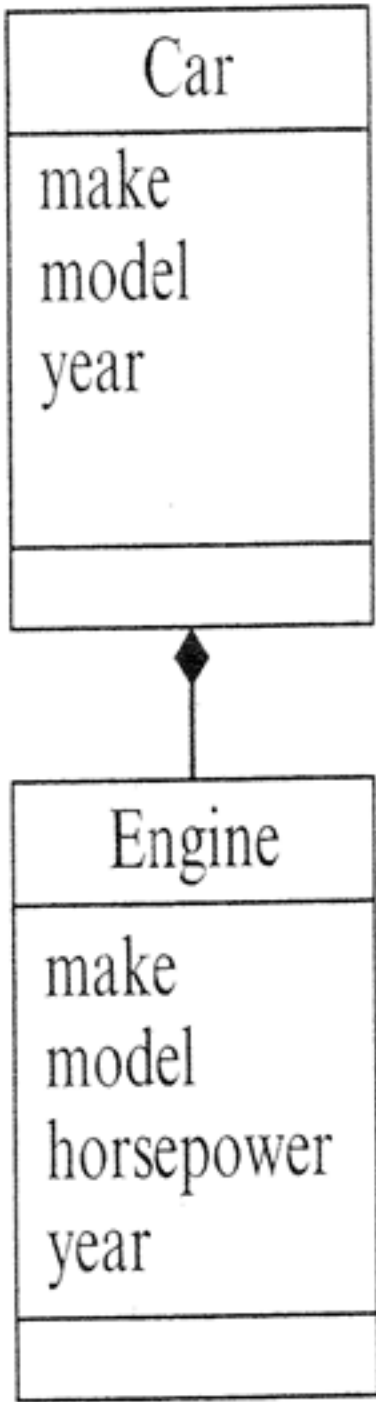
2



C

2

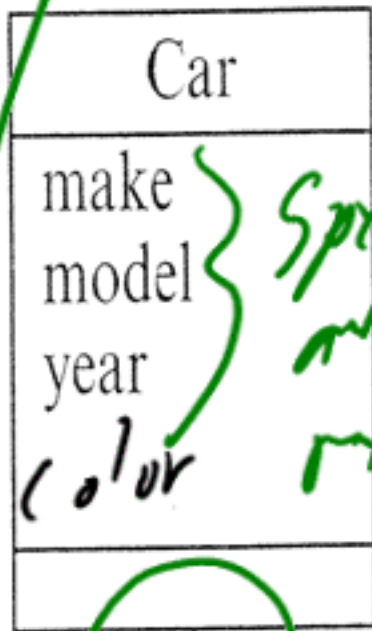
Which model is right?



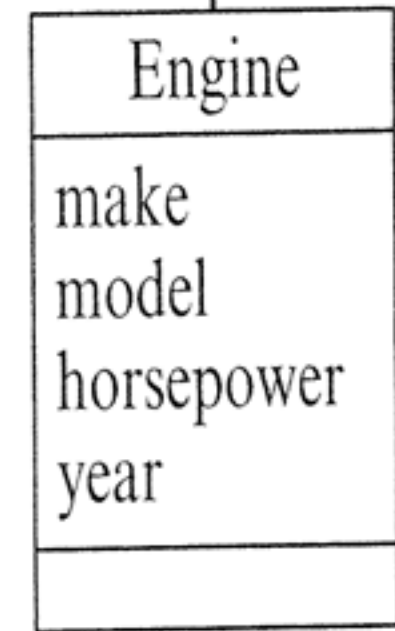
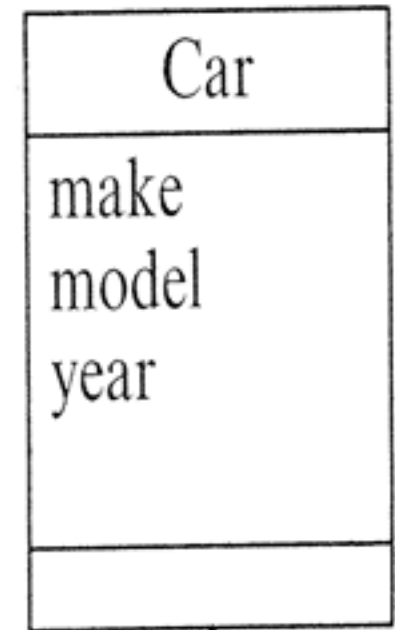
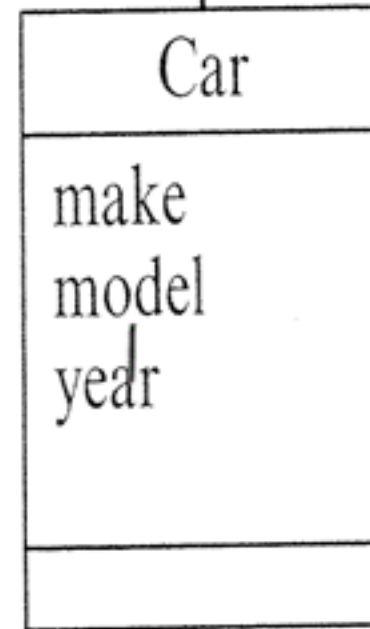
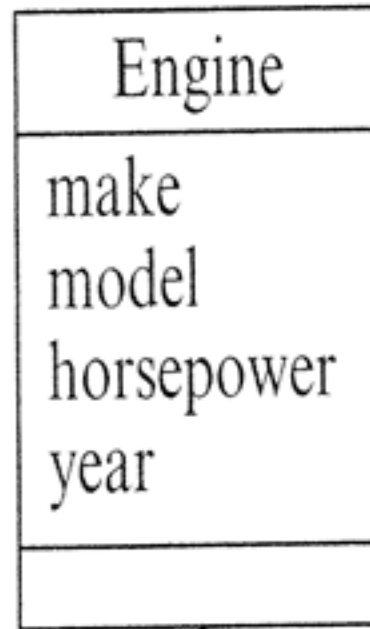
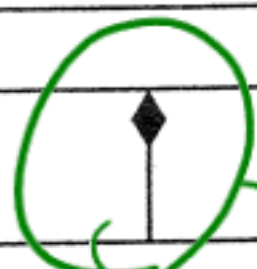
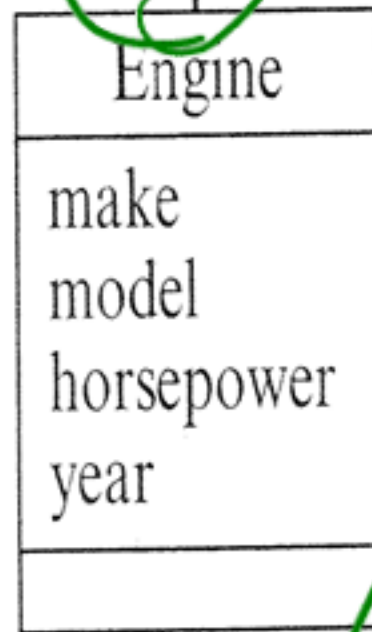
What's wrong?

inheritance
not inheritance
other relationship

Which model is right?



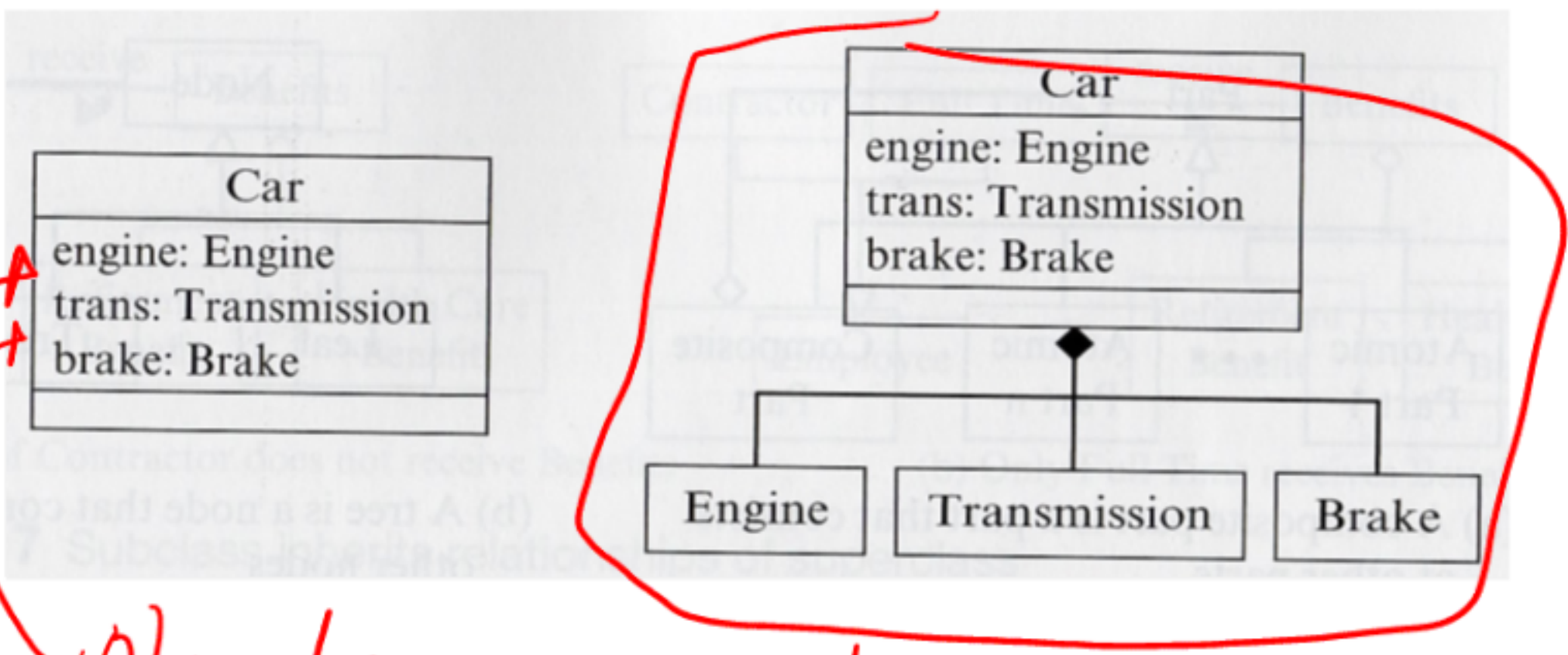
Specific attributes of car.



Car has an Engine.

What is wrong with these

UML models?

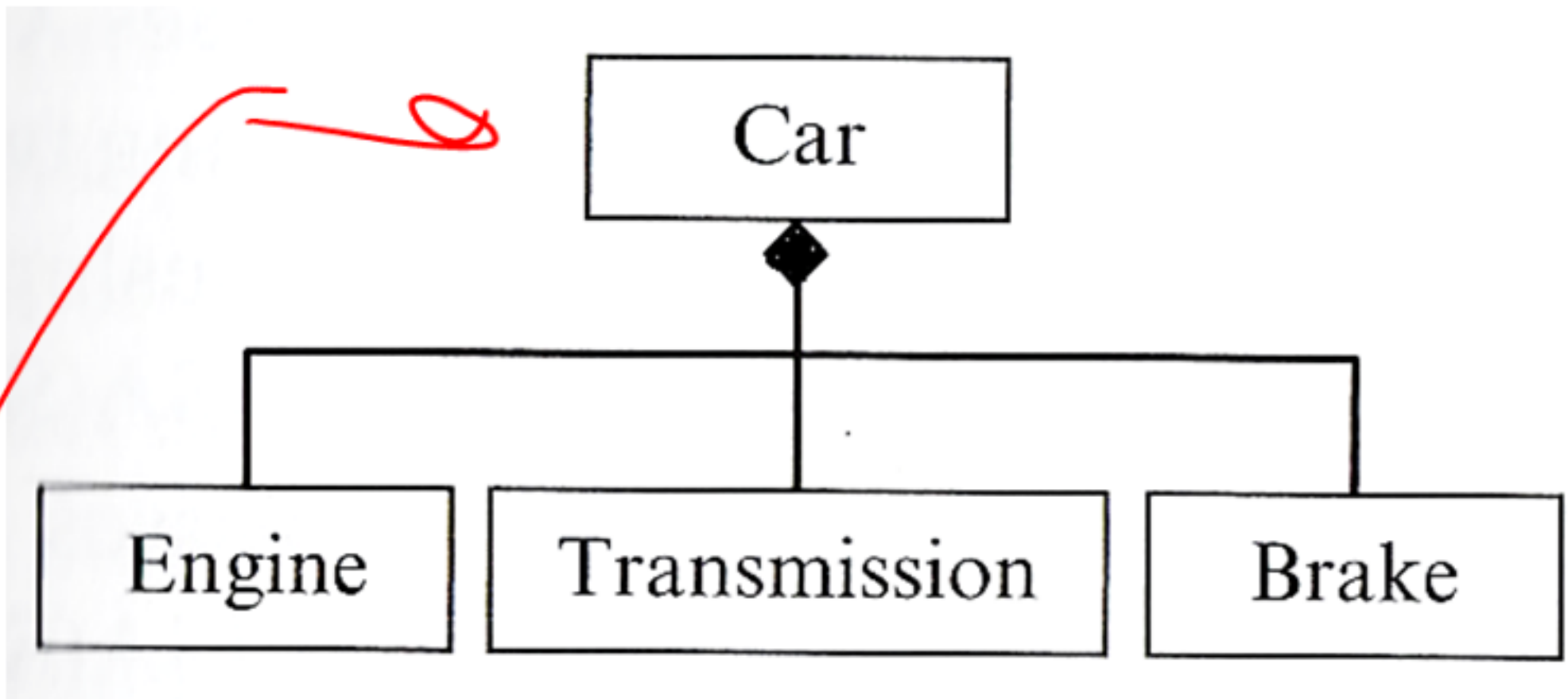


objects,
not attributes
Attributes
make an object
input.

Same problem
except I mean
trans and brake
are both objects
and attributes



The correct model.



Car has an engine,
Transmission, and a brake
on it.

How many

Class Diagram

- Captures the Vocabulary of a System

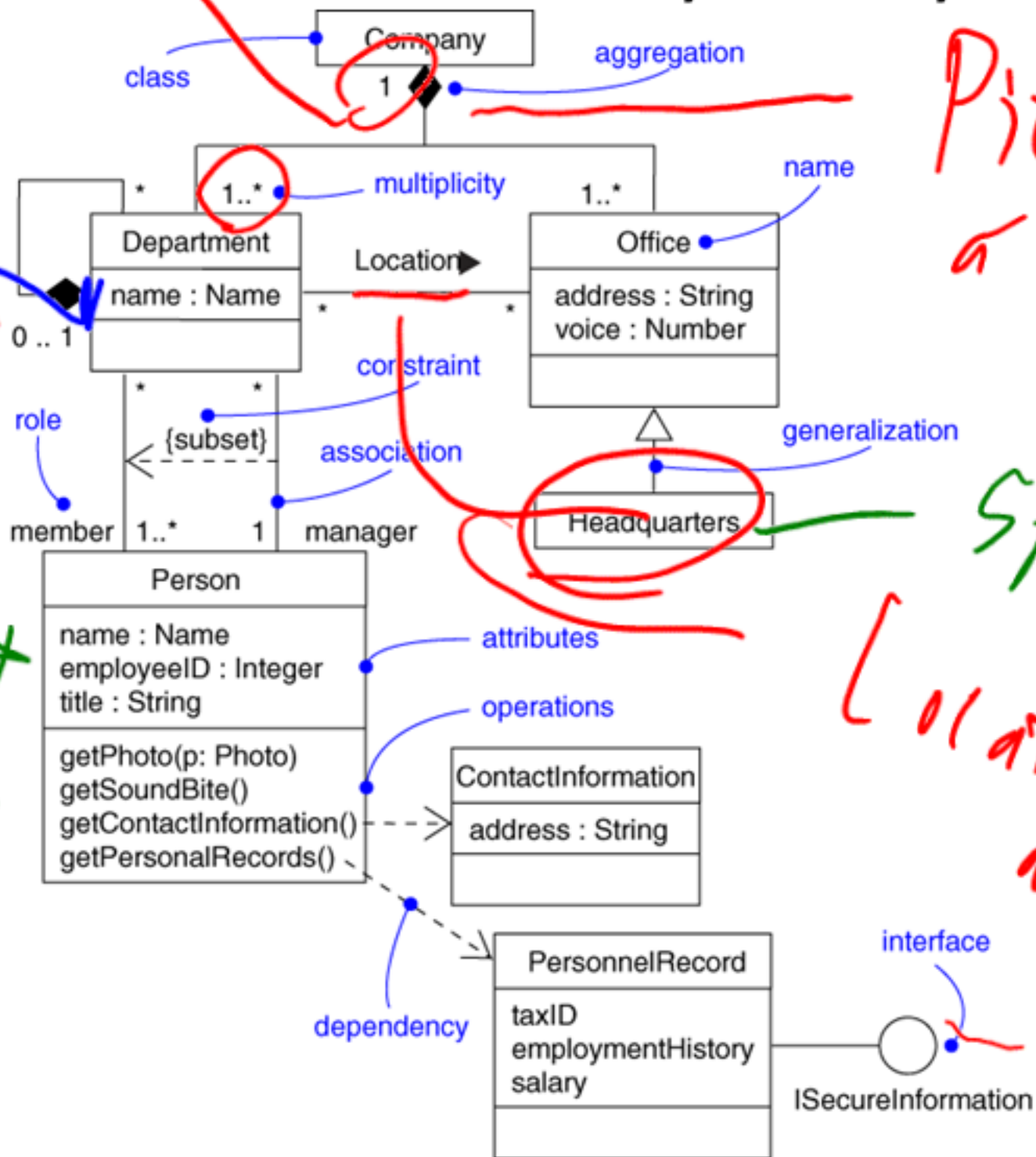
Makes Department Unique.

People

Pieces of a Company

Specialized

Located in an office



Model Example

- A software system is to be constructed which will allow a user to keep track of bank accounts stored within a bank. Each account shall have an initial deposit value as well as a unique account number. The bank shall keep track of accounts as well as the total assets of the bank. Through a user interface, accounts may be created, closed, deposited into, withdrawn from, and have inquiries performed against them. The user interface may also read in a file which contains a set of bank transactions.

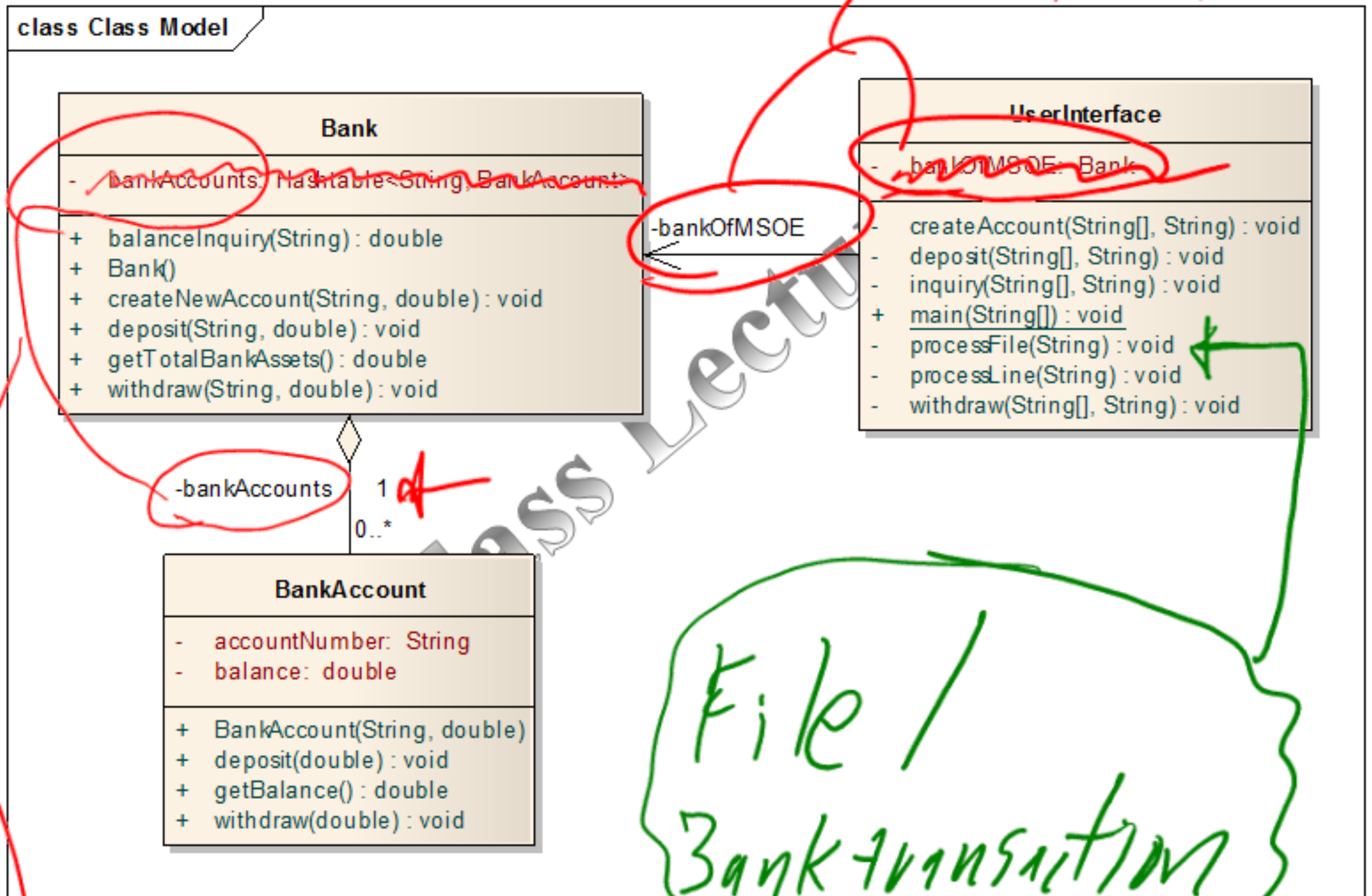
Candidate Objects

Domain Model

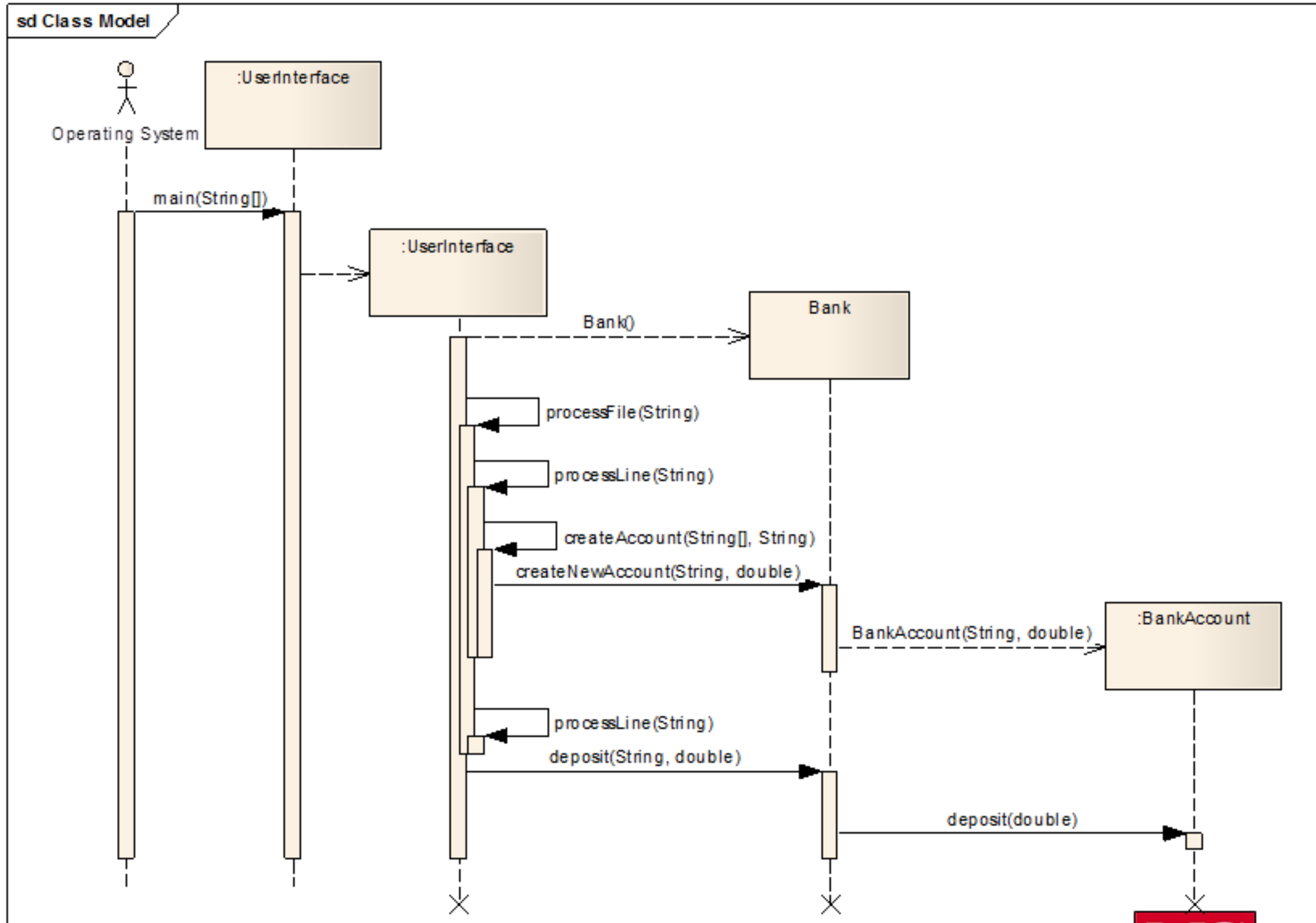
Domain Review Model Checklist

- Does the model contain the most important classes in the application domain? ✓
- Does the model show all important application domain relationships? ✓
- Does each relationship in the domain model correctly represent the real world relationship? ✓
- Are there any multiplicity constraints missing or incorrectly specified? ✓
- Does the model contain any design or implementation classes? ✓ *File?*
- Does every class in the domain model show all the important attributes? ✓
- Is the naming of each class, attribute, or relationship appropriate and easy to understand? ✓
- Are there any classes used as attribute types? ✓
- Do any of the classes show operations? ✓

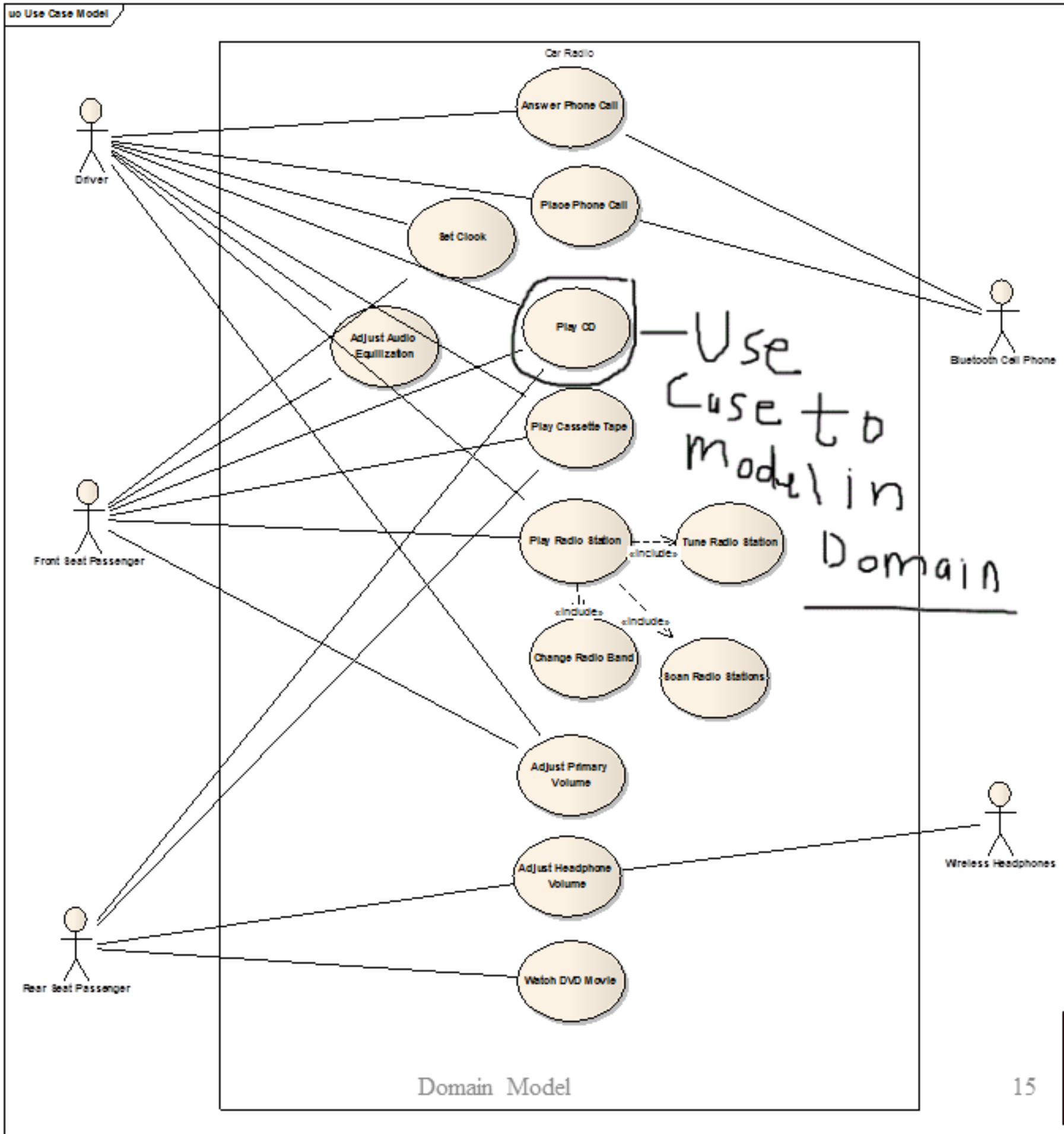
Potential Lab Solution

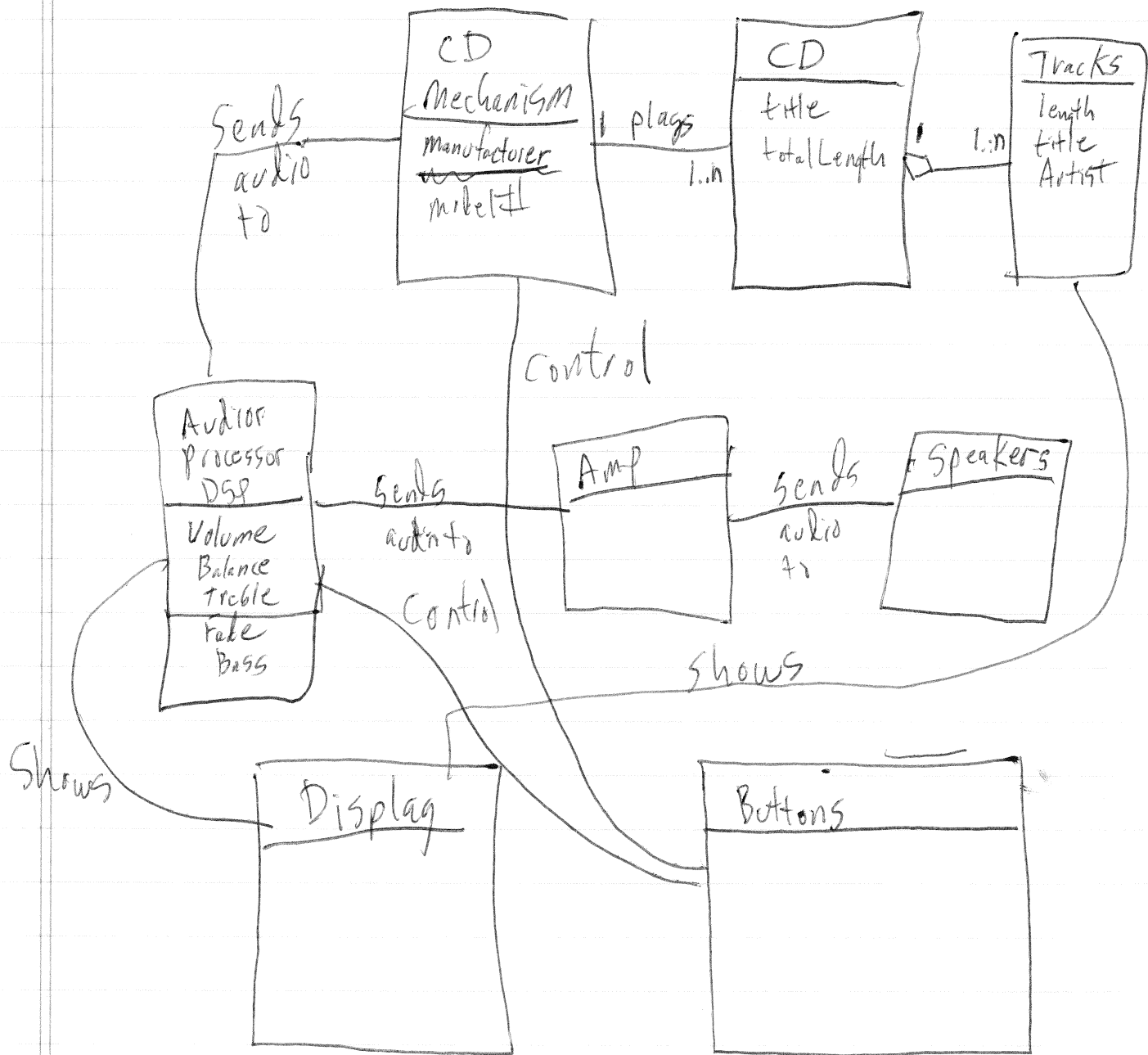


Partial Sequence Diagram



Domain Model Problem





UML

