CS489:
Applied Software Development

Lesson 2
Software Requirements and Domain Modeling

Wholeness

- This lesson gives an overview of the Software Development process, with a focus on how to identify requirements and performing Domain modeling.
- A Software development project typically starts with identifying the Requirements and then creating the Solution (domain) model using Object-Oriented Analysis and Design (OOAD) techniques.

Wholeness cont'd

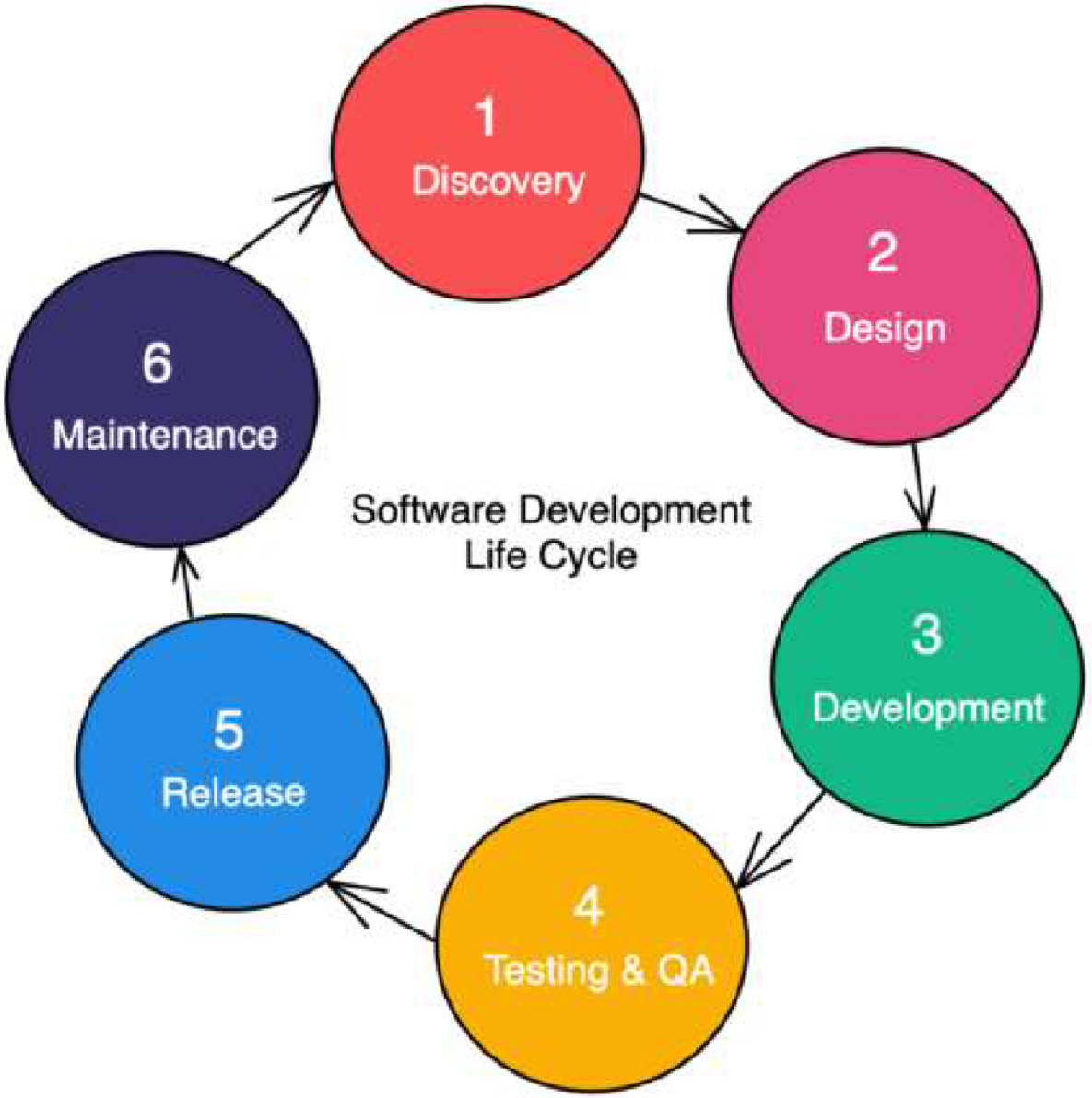
- The goal is to be able to identify Software Requirements from a given Problem Statement/description, perform Object-Oriented Analysis and Design and create a suitable Solution model – Domain modeling.
- Science of Consciousness: Action leads to Achievement. *Achievement leads to Fulfillment.*

Software Development Process

Requirements –> Analysis –> Design –> Implementation –> Testing –> Deployment, Operation & Maintenance

Software Development Life-cycle

- Typically, a Software Development project will go through following activities in its lifecycle:



Software Development Methodology

- Software Development methodology is the process that a project team follows in creating software product. The methodology a team adopts/practices is what determines how the various software development life-cycle activities will be performed/executed.
- The following methodologies can be used:
  - Agile: Scrum, Extreme Programming (XP) etc.
  - Rational Unified Process (RUP)
  - Waterfall
  - Kanban, Lean

Software Requirements Discovery

- Consider the following Problem statement/description:

A City's Department of Public Libraries has hired you to update their library record keeping operations.

The department currently owns and runs a network of public libraries at three different locations within the city. Currently the libraries share an electronic card catalog that contains information such as author, title, publisher, description and location of all of the books in the libraries. All the library member information and book check-in and checkout information, however, is still kept on paper, at each location. This system was previously workable, because the Libraries had only a few registered members. Due to the increasing city population, the Library now needs to automate the check-in and checkout system.

Software Requirements Discovery

- City Library Problem statement cont'd:

The new system will have a web-based application to allow librarians to check-in and checkout books. The system should also maintain information about each publisher, including their head-office address and main telephone number, to enable the libraries order new supplies of books. All the books in the library are classified under a subject (such as Science, Arts, Medicine, Poetry, Engineering, Music, History etc.), to enable members find books that are pertaining to a subject which they are interested in. All books in the library have a unique bookid, in addition to the book's unique ISBN number. The books in the library are ordered on the shelves by their bookid. The new system must allow registered library members to search through the electronic card catalog to find the bookid of the desired book.

Software Requirements Discovery

- City Library Problem statement cont'd:

The system will either run on a server hosted by the library or hosted in a cloud service. Librarians and library members will be able to gain access into and use the system through a web-browser interface. However, only librarians are able to check-in and checkout books.

The system will retain information on all library members, including their addresses, phone numbers and their fees payment options (in form of the debit or credit card information, to be used to pay any fees owed). Only valid identifiable residents of the City of Fairfield can become library members. To be registered as a Library member, a valid City of Fairfield-issued identification card is required to be presented to the Librarian.

Software Requirements Discovery

- City Library Problem statement cont'd:

Standard members can check-out books for a maximum of 21 days. If a standard member returns a book later than 21 days, then he/she has to pay an overdue fee of 25 cents per day. Member who are staff of the Library, including the Librarians, can also checkout books for a maximum of 21 days, but pay an overdue fee of 10 cents per day. Members who are senior citizens (i.e. those of age 70 or older) can checkout books for a maximum of 42 days, and they pay only 5 cents per day for every book returned late.

The system will keep track of the amount of money that each library member owes the library. No member will be able to checkout a book if they have another book overdue or owe a fee greater than 100 cents.

Software Requirements

- **Exercise**: Identify the software requirements from the above given problem statement for the City Library system, and specify them in a list of short statements.

15 minutes

For example:

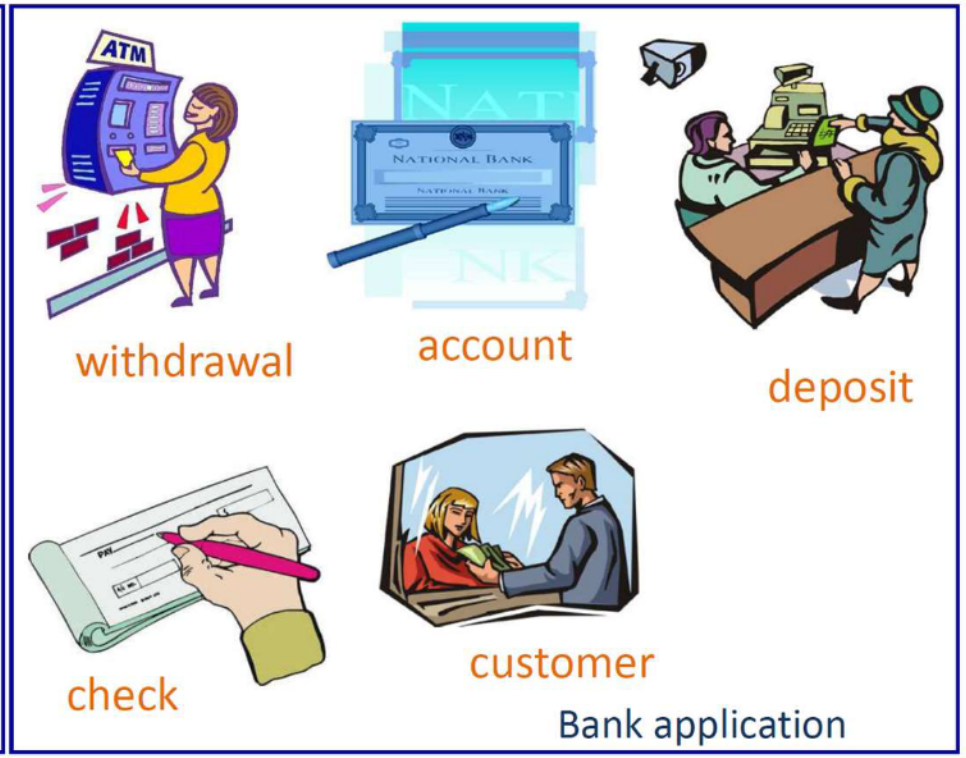
1. The system should allow users to be able to Sign-in
2. The system should maintain data about each book in the library
3. The system should allow library member to search for books
4. …

Systems Analysis and Design

- After the requirements have been identified, the next step in the software development process is to perform Analysis to achieve a good understanding of those requirements.
- Based on the analysis, a solution design is then created.
- In modern software development practice we apply Object-oriented Analysis and Design

Object-Oriented Analysis & Design

- Object-Oriented Analysis and Design (OOAD) is the technique by which the software developer identifies the essential **Entity** objects in the problem domain, from which the software solution will be formulated.



Domain-Driven Design

- Domain-Driven Design (DDD) provides a set of principles and patterns that the developer can use to create an elegant object-oriented solution model known as the Domain model.

Domain-Driven Design

- In applying Domain-Driven Design (DDD) the software developer creates the Domain model by identifying and defining Domain Entities, Value objects, behaviors, relationships and business rules found in the real-world domain.
- The domain model is a conceptual model that can then be represented using a UML class diagram.

Overlap: DDD often uses object-oriented principles and techniques to implement the domain model. For example, entities and value objects in DDD are typically implemented as classes in an object-oriented language.

Complementary: While OOAD provides the tools and techniques for designing object-oriented systems, DDD provides a strategic approach to ensure that the design is deeply rooted in the domain knowledge and business requirements.

In summary, DDD can be seen as an approach that builds on the principles of OOAD but with a strong emphasis on the domain and business logic. They are complementary, with DDD providing the strategic direction and OOAD offering the tactical tools for implementation.

Object

- An Object is a concept or thing in the problem domain

Customer

Destination

Date

Travel System

Luggage

Creditcard

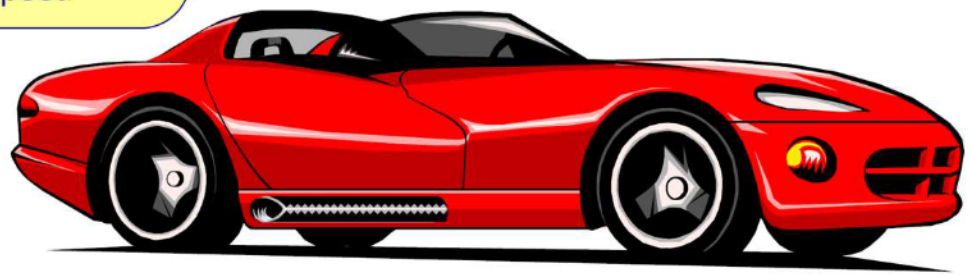
Flight

Rental car

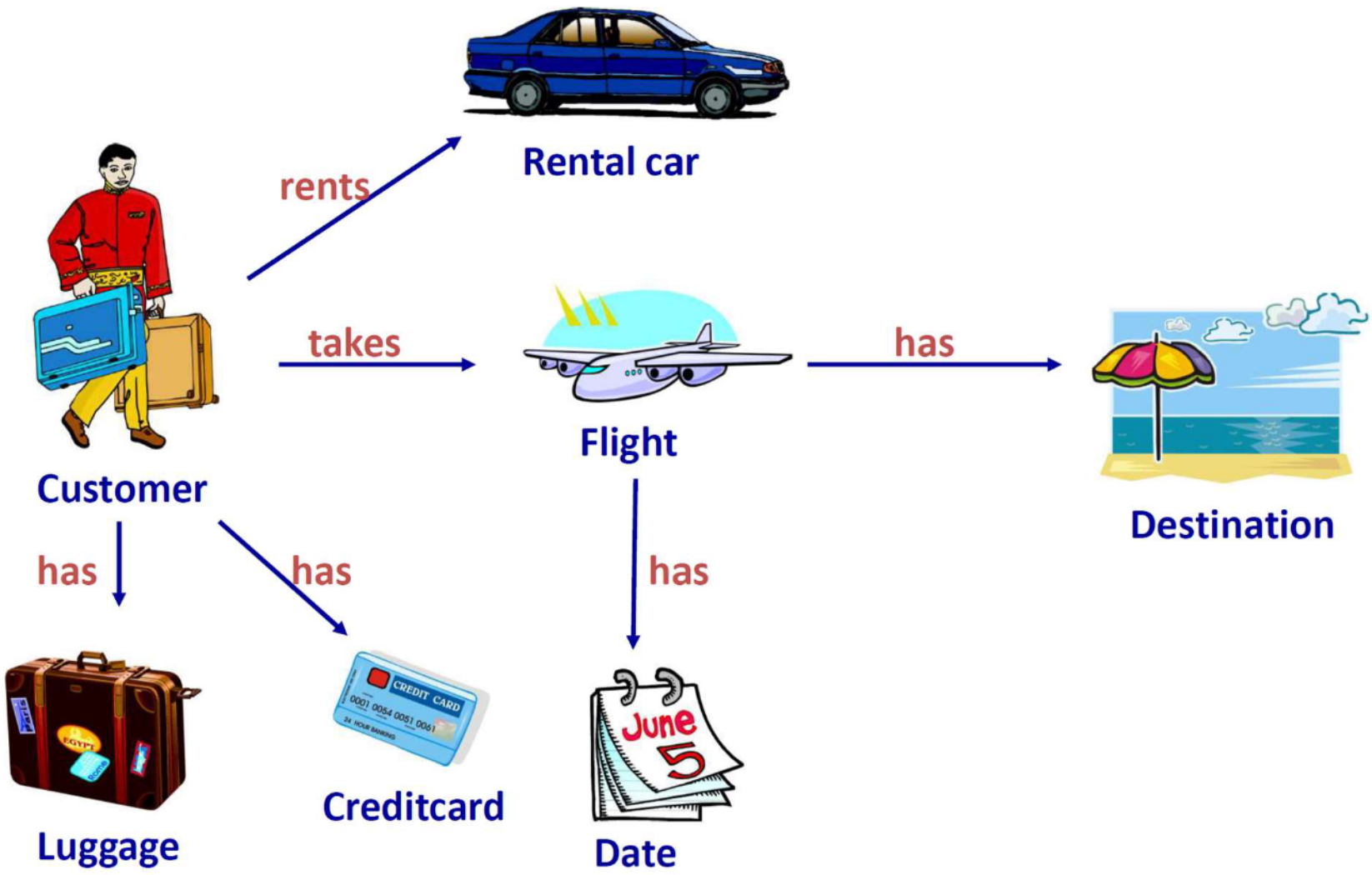
An Object has…

Data
- Color: red
- Price: $12000
- Year: 2001

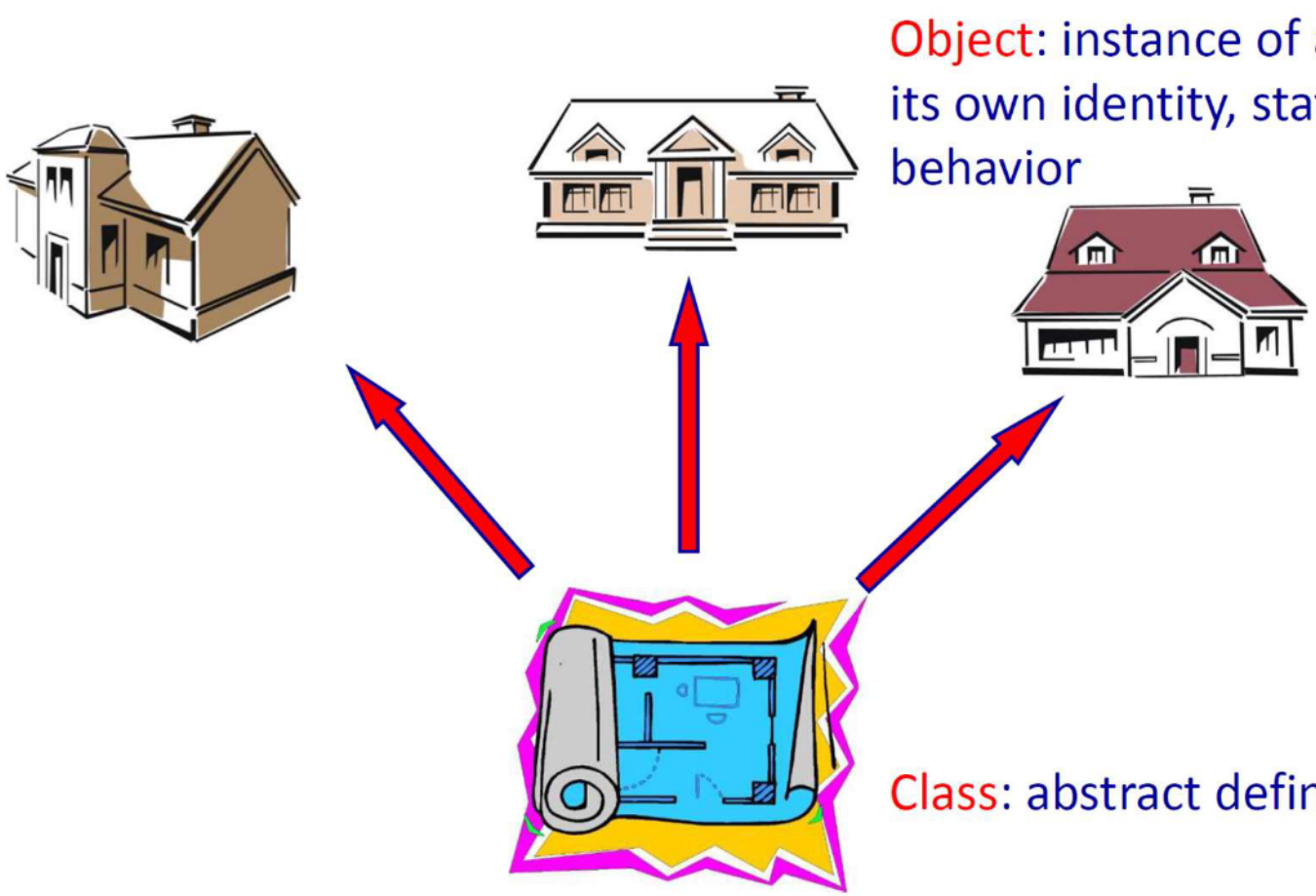
Behavior
- Drive
- Brake
- Turn
- Increase speed
- Decrease speed



Object relationships



Class



Object: instance of a class with its own identity, state and behavior

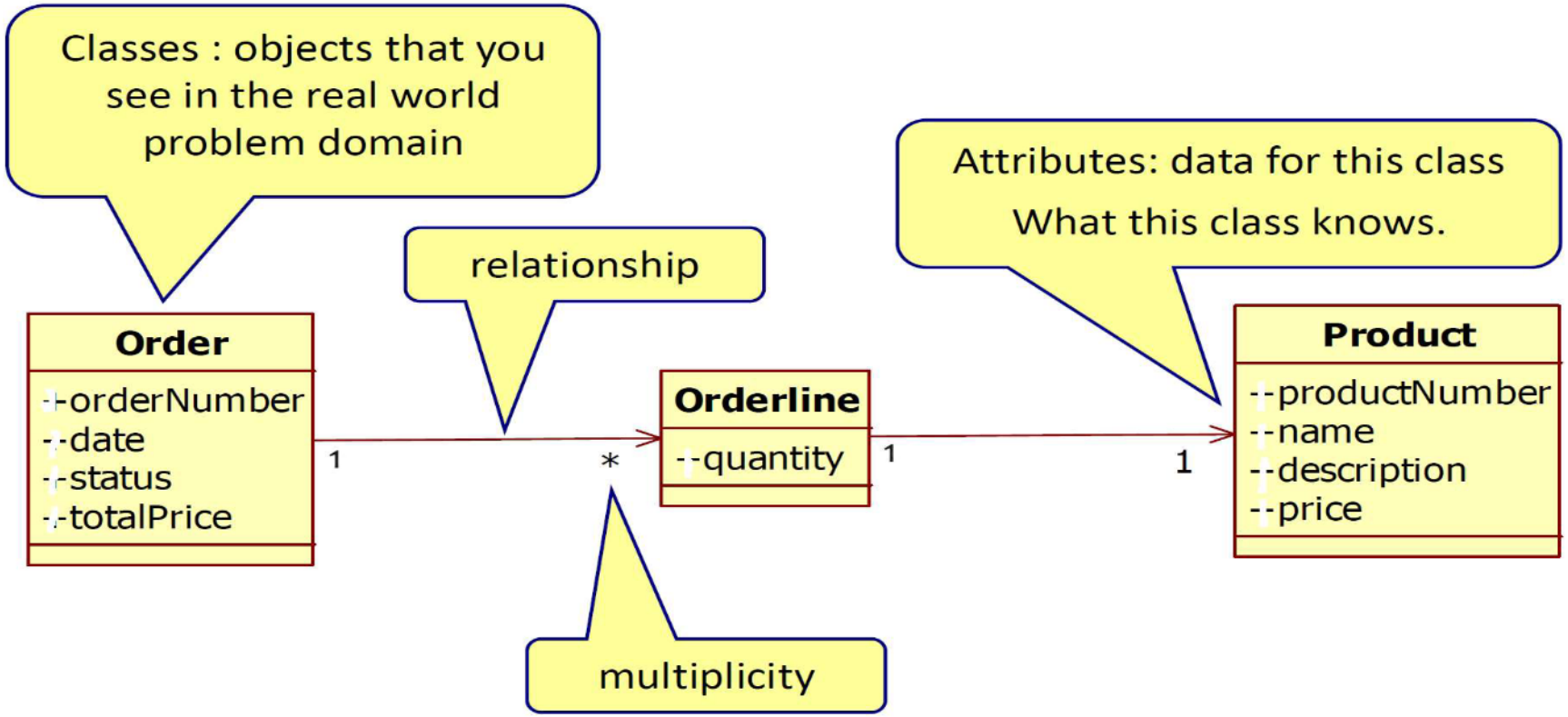
Class: abstract definition of an object

Identifying Objects

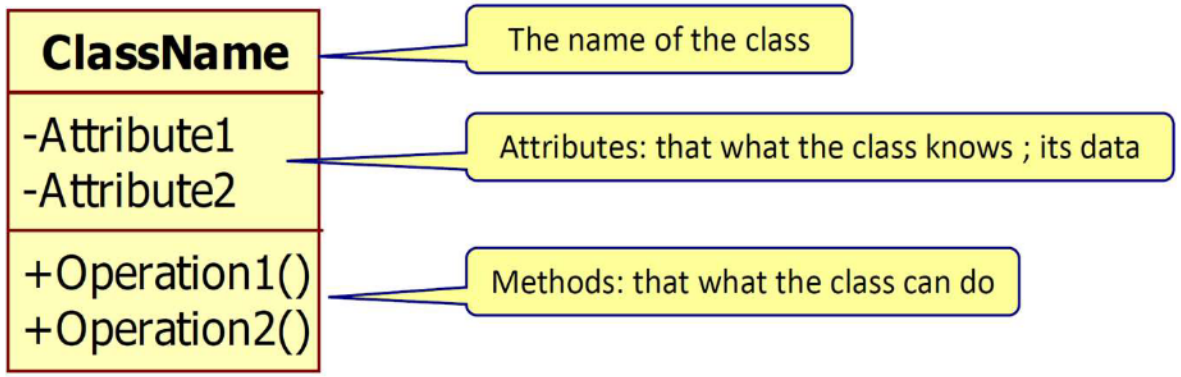
- Identify the concepts and things in the problem domain.
- Apply None/Verbs analysis technique
  - Nouns will become classes or attributes
  - Verbs will become methods

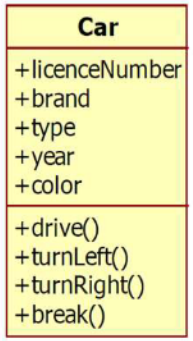
Domain model class diagram

- The domain model is created as a UML class diagram. It represents the structure of the software solution, showing the classes, their attributes and the relationships



Class in a UML class diagram





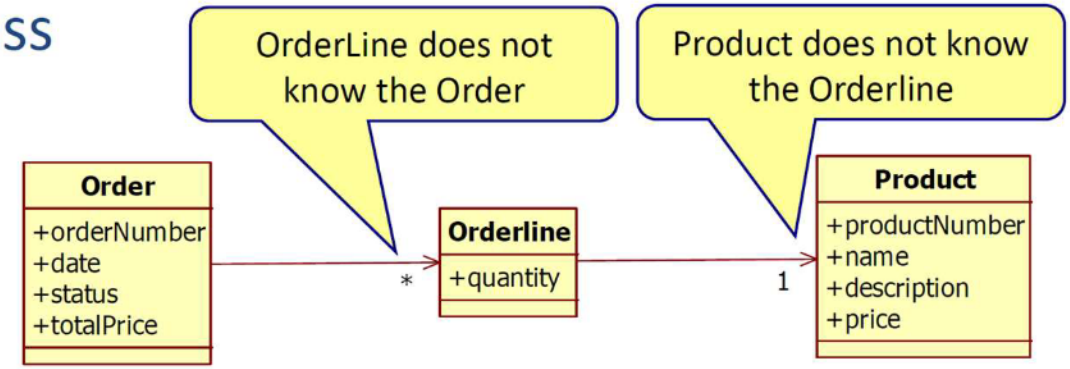
Relationships in domain model class diagram

- Association

  Defines the relationship between 2 classes

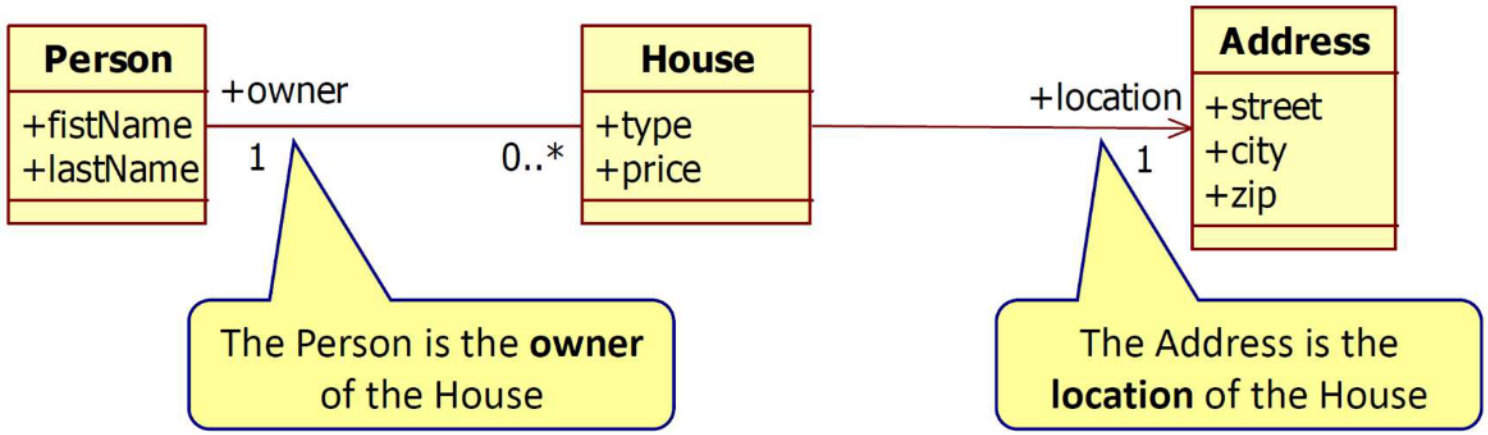
  - Bi-directional: both classes are aware of each other

  - Uni-directional: only one class is aware of the other class



Relationships in domain model class diagram

- Association Role
  - Indicates the role played by the class attached to the end of the association path



Relationships in domain model class diagram

- Multiplicity
  - 0
  - 1
  - 0..1
  - 0..*
  - 1..*
  - *
  - 7
  - 5..7
  - 2,4

Zero or one

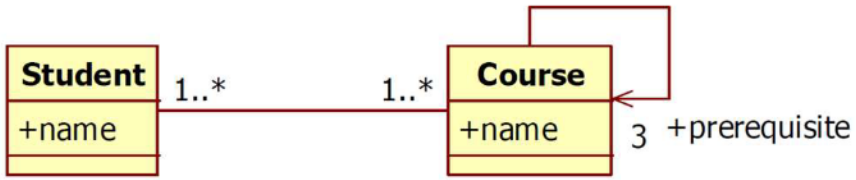
Zero or more

One or more

Zero or more

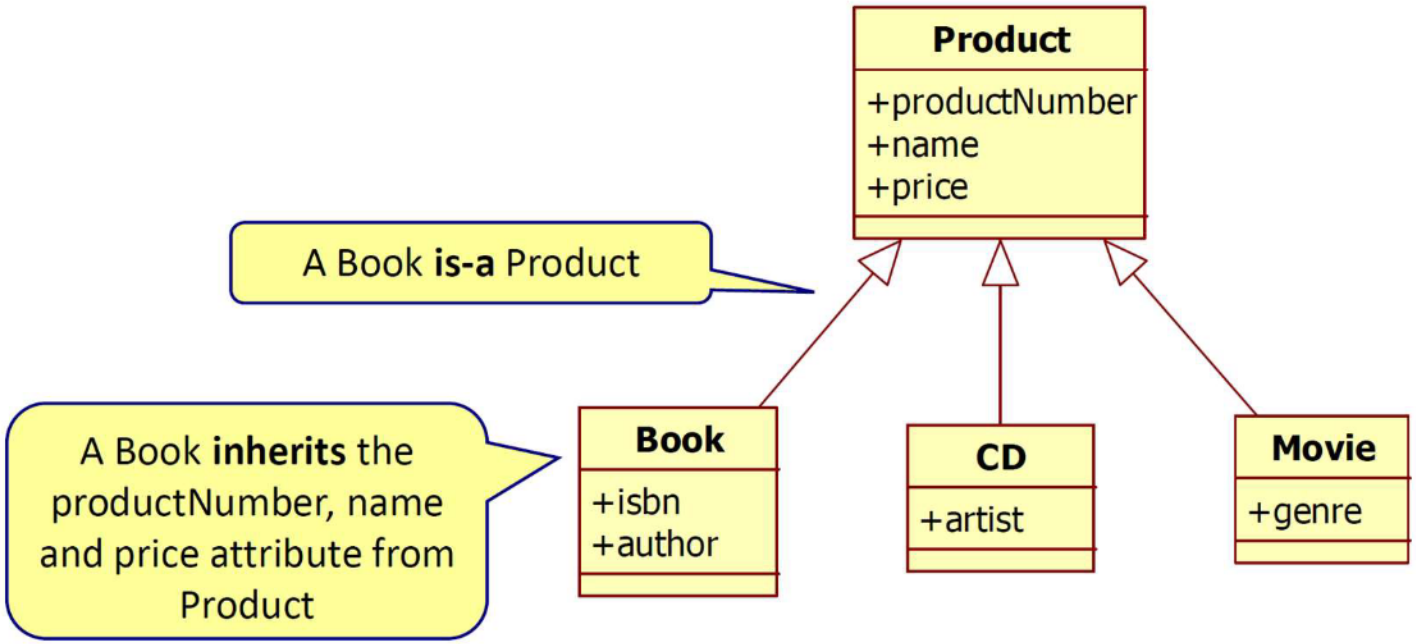
Specified range

Two or four



Relationships in domain model class diagram

- Inheritance
  - Is-a relationship



Finding classes from the problem statement

- Library members can check-out books

LibraryMember

CheckoutRecord

Book

- Library members can reserve books

Reservation

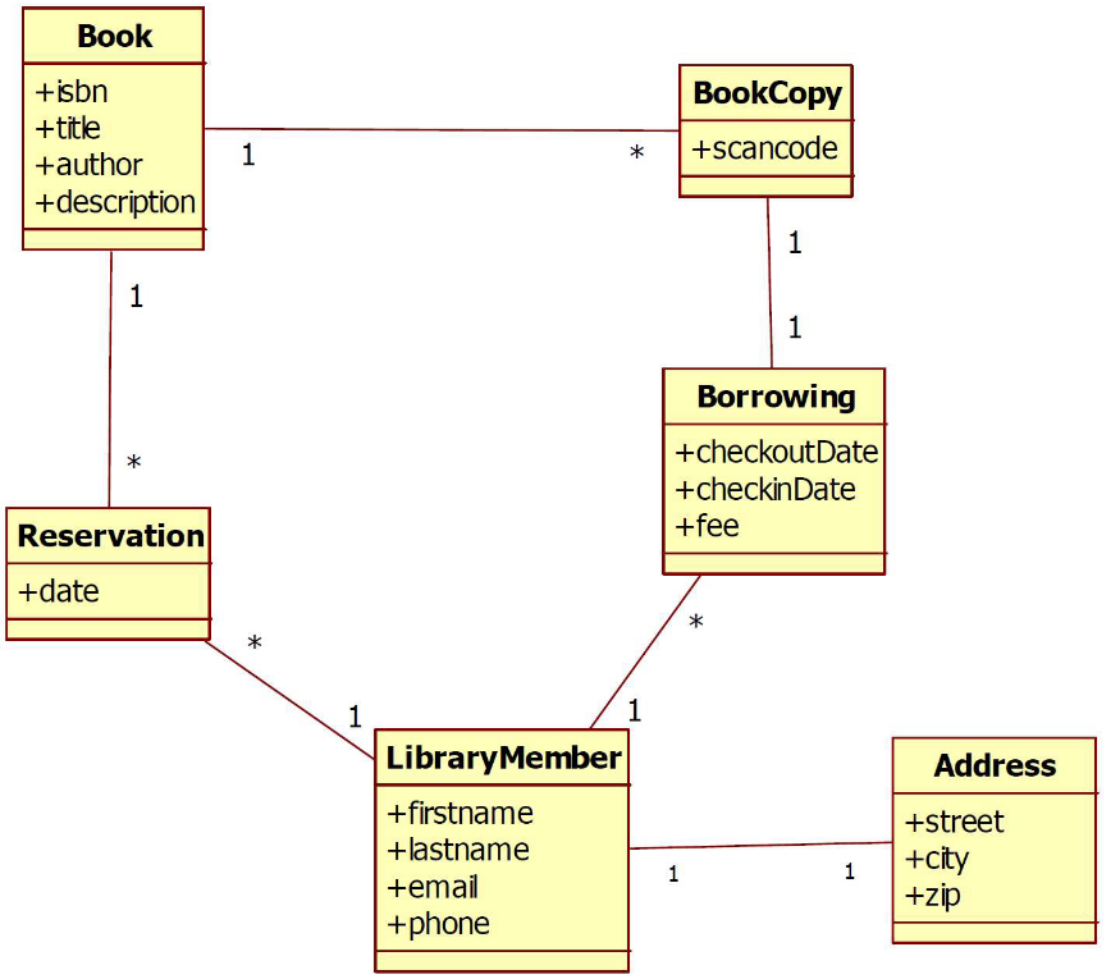
- We can have multiple copies of one book

BookCopy

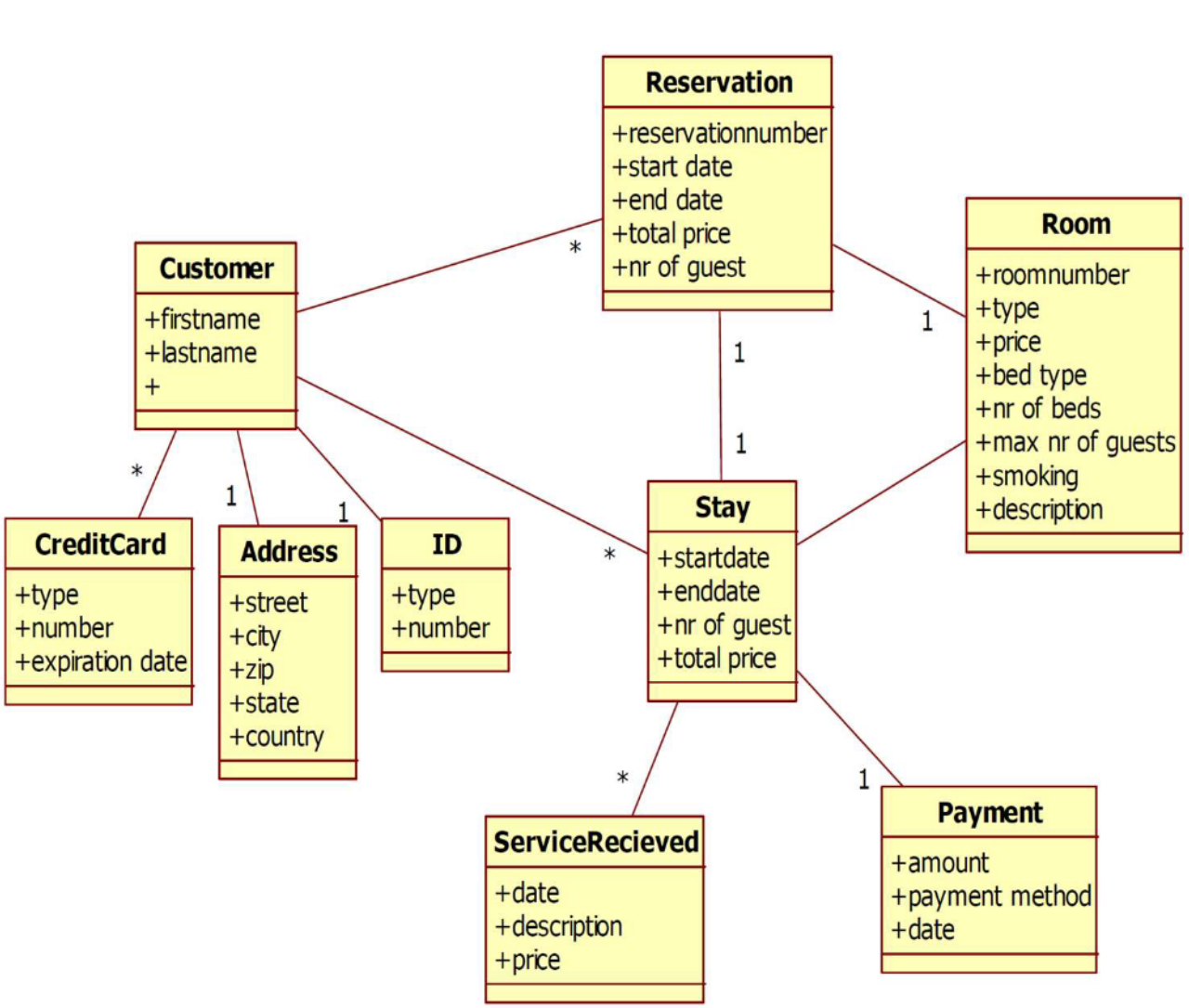
- Library members live at a certain address

Address

Library system domain model



Hotel reservation system domain model



Domain modeling

- **Exercise**:

  Perform analysis of the requirements from the given problem statement for the City Library system, and create a domain model class diagram.

CS489:
Applied Software Development