# Object Oriented Software Engineering

## UML Diagrams: System Sequence Diagram & Domain Model

**Presented By:**

**T.A. Asmaa Hammad El-sayed**

**E-mail: eng.asmaa134@gmail.com**

# Agenda




**System Sequence Diagram**


**Domain Model**

# System Sequence Diagram

# System Sequence Diagrams

- A system sequence diagram shows the interaction between an actor and the system for one use case scenario.
- For a use case scenario, an SSD shows:
  - The System (as a black box)


  - The external actors that interact with System
    - The messages into and out of the system
    - The sequence (order) in which the messages occur
  - Develop SSDs for the main success scenario of a selected use case, then frequent or complex alternative scenarios

# System Sequence Diagrams

- How to construct an SSD from a use case:
  - Draw a rectangle representing the system. Label the rectangle and draw a vertical dashed line (lifeline) under it.
  - At the left, draw **rectangle shape** or a **stick figure** for each actor. Label it with the actor's name and draw a vertical dashed line (lifeline) under it.
  - For each system input, draw a message arrow from the actor's lifeline to the system's lifeline. Label it with the message name and parameters.
  - Confirm that the sequence of messages (from top to bottom) is correct.

# System Sequence Diagrams

- Message Formats in System Sequence Diagrams
  - The UML format for a message consists of a message name followed (in parentheses) by a parameter list.
  - All names begin with a lower-case letter.
  - There are no spaces in a name.
  - Upper-case letters separate the words within a name.
  - Names in the parameter list are separated by commas.
  - Start massage names with verb

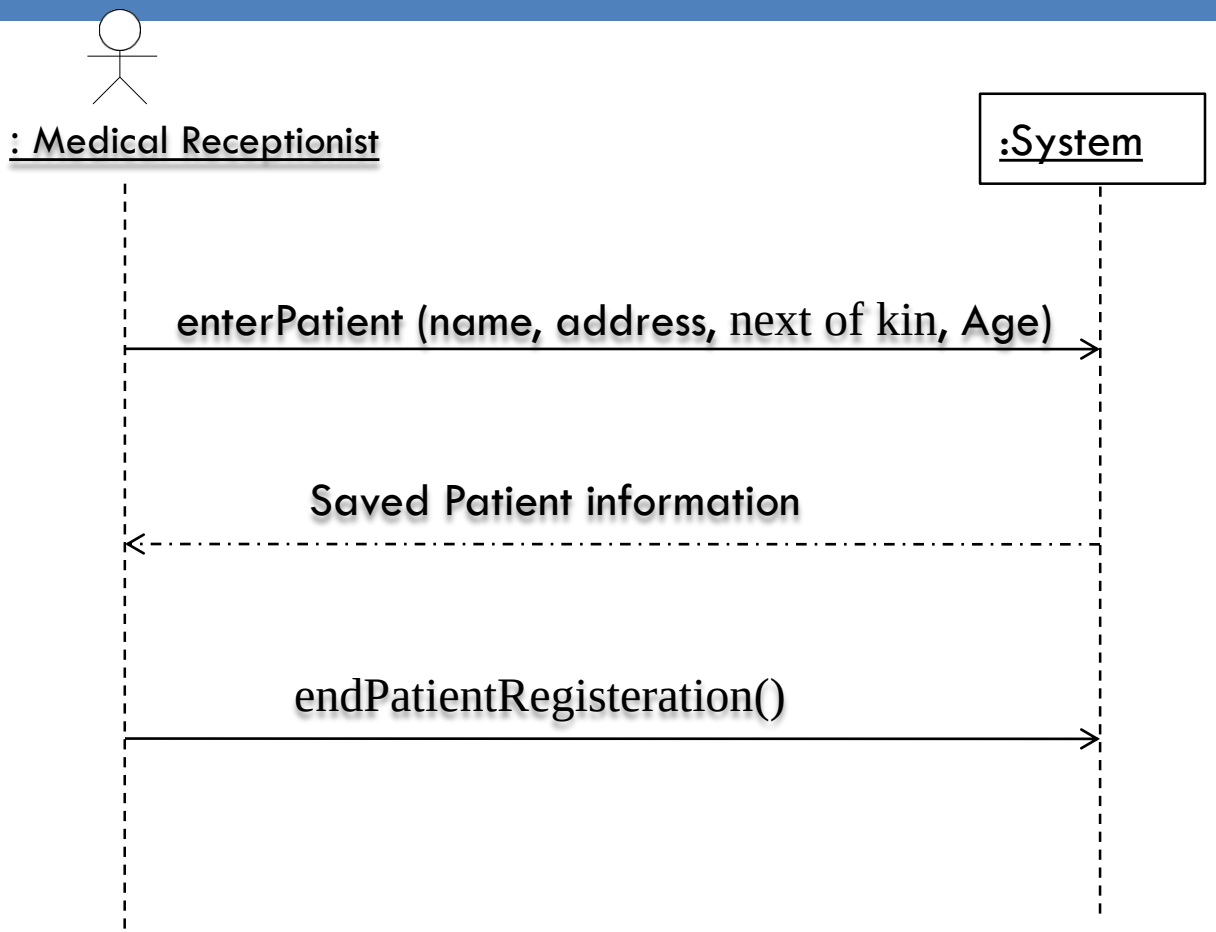
# Examples

# Example 1: SSD for Register Patient in MHC-PMS

**Draw the SSD for use case (register a Patient)?**

When a new patient attends a clinic, the medical receptionist asks the about his/her data which include name, age, address, and next of kin. A new record is created by a medical receptionist and personal information (name, address, age, etc.) is added to it.

# Example 1: SSD for Register Patient in MHC-PMS



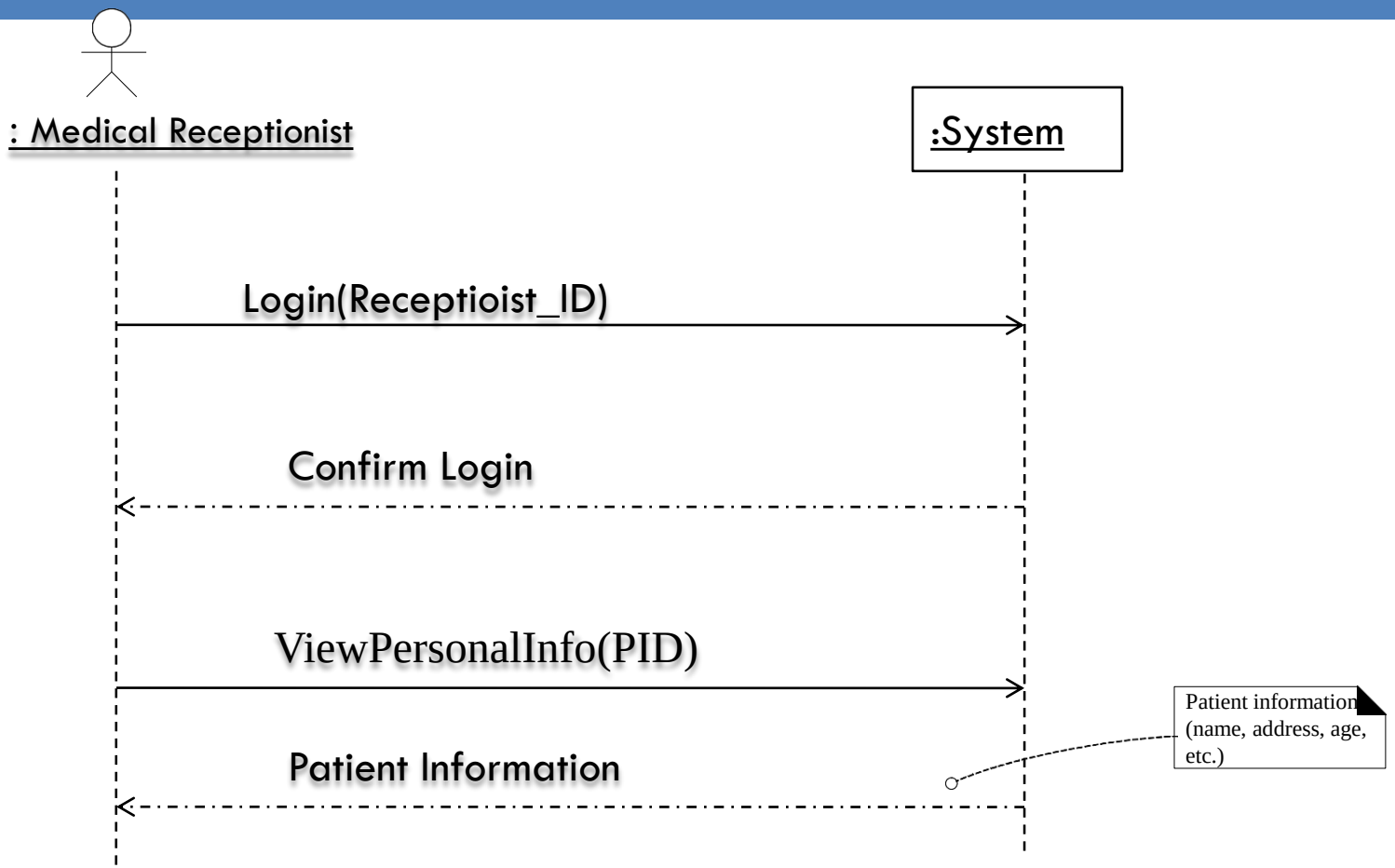
# Example 2: SSD for View personal info in MHC-PMS

**Draw the SSD for use case (View personal info )?**

**Main success scenario:** The medical receptionist logs on to the system to View Info about patient, supplying the receptionist's identifier to allow security checking . BY supplying the patient's identifier, PID , the database return the information required. Through the database, the system checks that the user is authorized for this action. If authorized, the patient information is returned and a form on the user's screen.

**Alternative scenario:** If authorization fails, then an error message is returned and the system asks the user to enter another identifier.

# Example 2: SSD for View personal info in MHC-PMS



# Example 3: SSD for Collecting medical history in MHC-PMS

**Draw the SSD for use case (Collecting medical history)?**

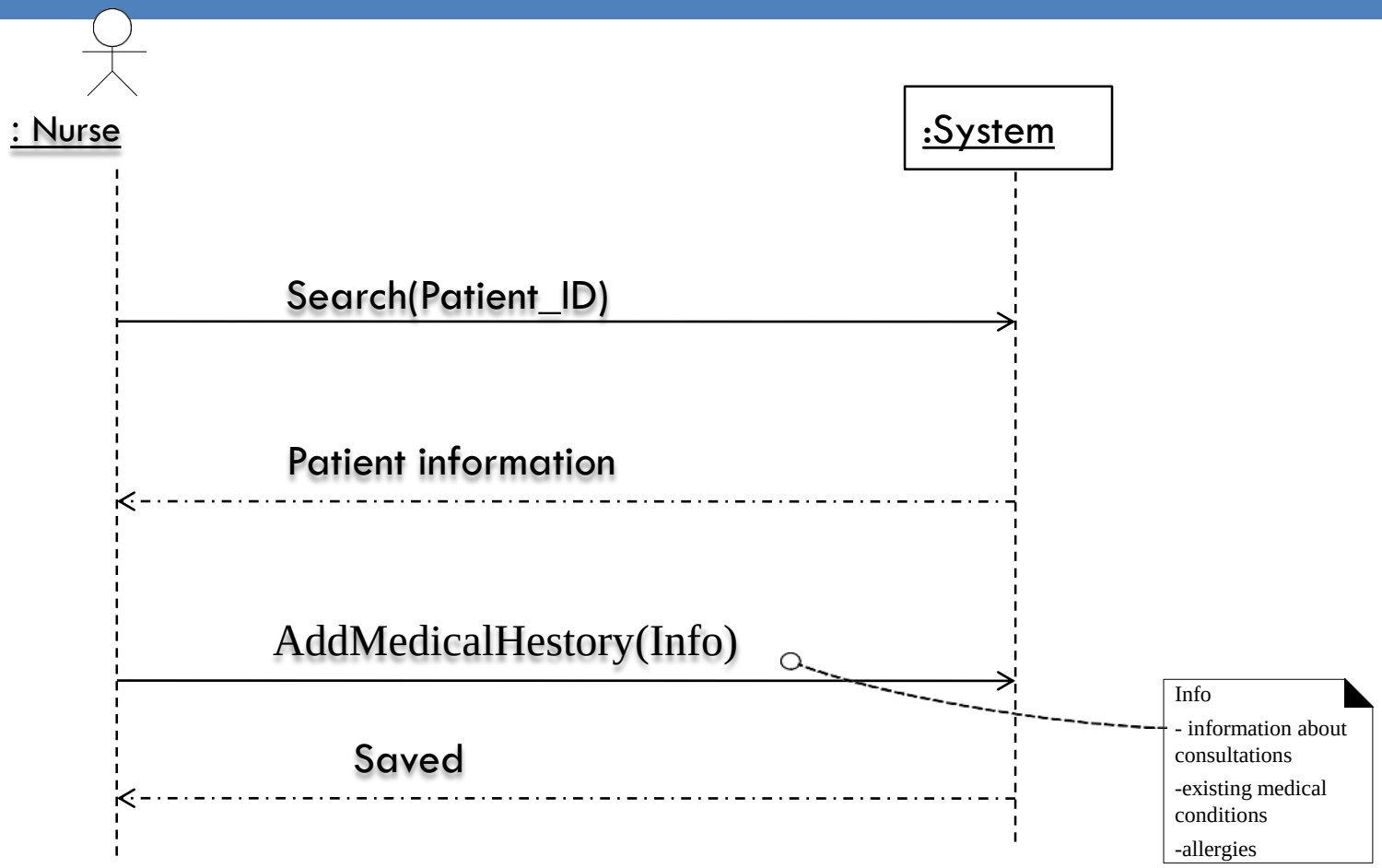
**Main Success Story**

1.The nurse searches for the patient by Patient ID.

2.The nurse chooses the menu option to add medical history.

3.The nurse then follows a series of prompts from the system to enter information about consultations elsewhere on mental health problems (free text input), existing medical conditions (nurse selects conditions from menu), medication currently taken (selected from menu), allergies (free text), and home life (form).

# Example 3: SSD for Collecting medical history in MHC-PMS



# Domain Model

# Domain model



- **What?**
  - Domain model helps us to identify the relevant concepts and ideas of a domain
  - Illustrates meaningful conceptual classes in problem domain
  - Represents real world concepts, not software components
  - Software oriented class diagrams will be developed later, during design

# Domain model(cont.)



- **Steps to create a Domain Model**
  - ✓ Identify candidate conceptual classes.
  - ✓ Draw them in a UML domain model.
  - ✓ Add associations necessary to record the relationships that must be retained.
  - ✓ Add attributes necessary for information to be preserved.
- **No operations are defined in domain model**

# Conceptual Classes



- **To identify conceptual classes**
  - Use a conceptual class category list .
    - ✓ Make a list of all candidate conceptual classes
  - Identify noun phrases
    - ✓ Identify nouns and phrases in textual descriptions of a domain ( use cases, or other documents)
    - ✓ Use Cases" are also an excellent source for identifying conceptual classes.

# Conceptual Classes Category List

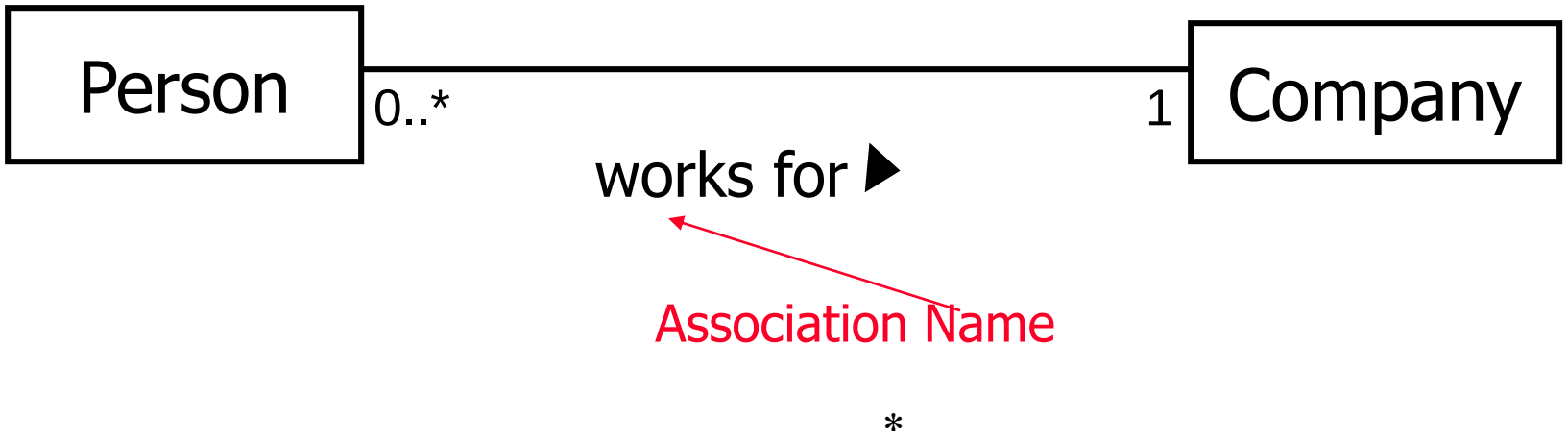


- **Physical objects**
  - Item, Airplane
- **Specifications, or descriptions of things**
  - ProductSpecification, FlightDescription
- **Places**
  - Store, Airport
- **Transactions**
  - Sale, Payment, Reservation
- **Roles of people**
  - Cashier, Pilot
- **Containers of other things**
  - Store, Airplane
- **Things in a container**
  - Item, Passenger
- **Catalogs**
  - ProductCatalog,
- **Organizations**
  - SalesDepartment, Airline
- **Events**
  - Meeting, Flight

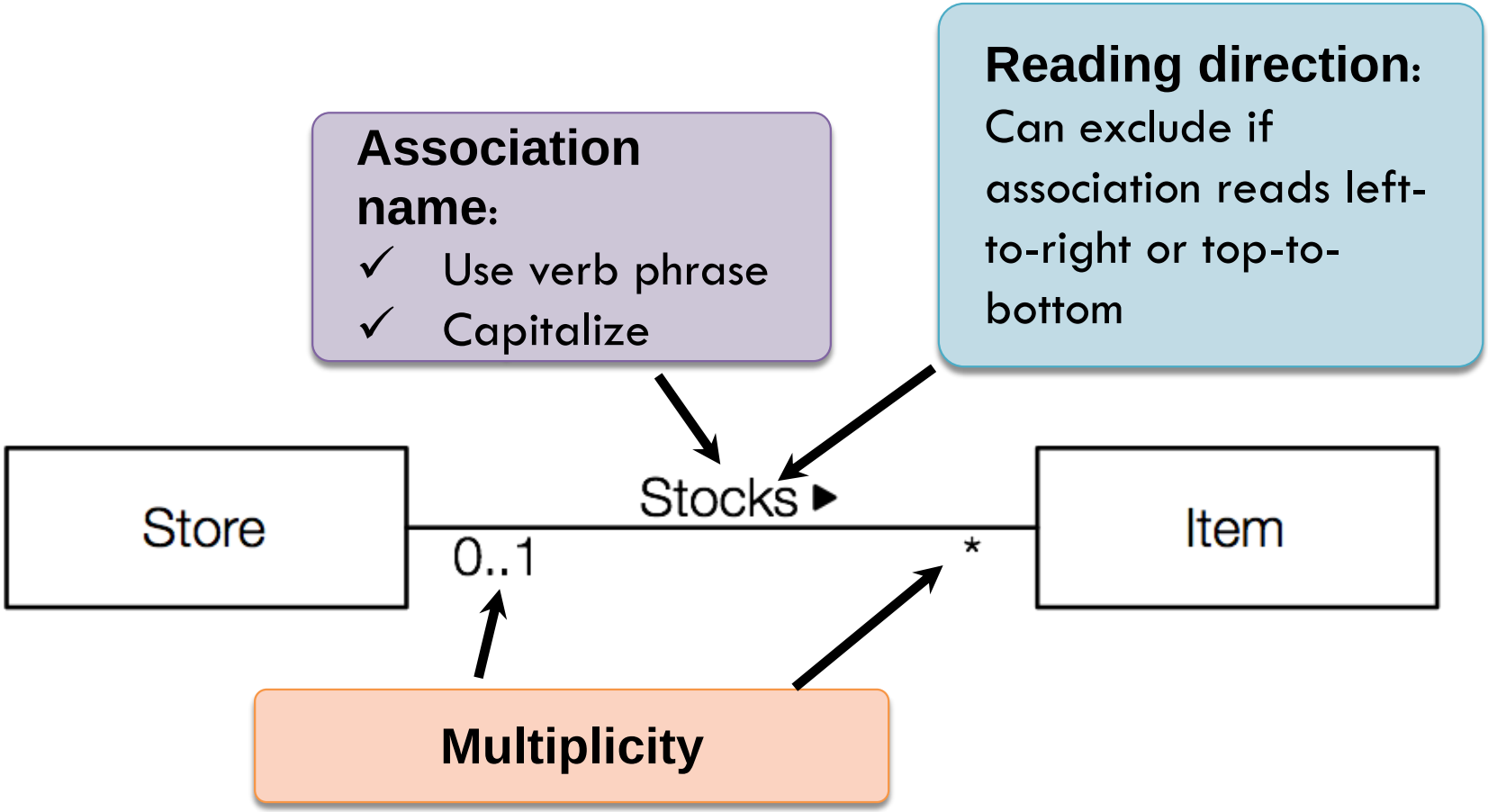
# Adding associations



- Associations
  - An association is a relationship between classes that indicates some meaningful and interesting connection.
  - **A link between two classes**



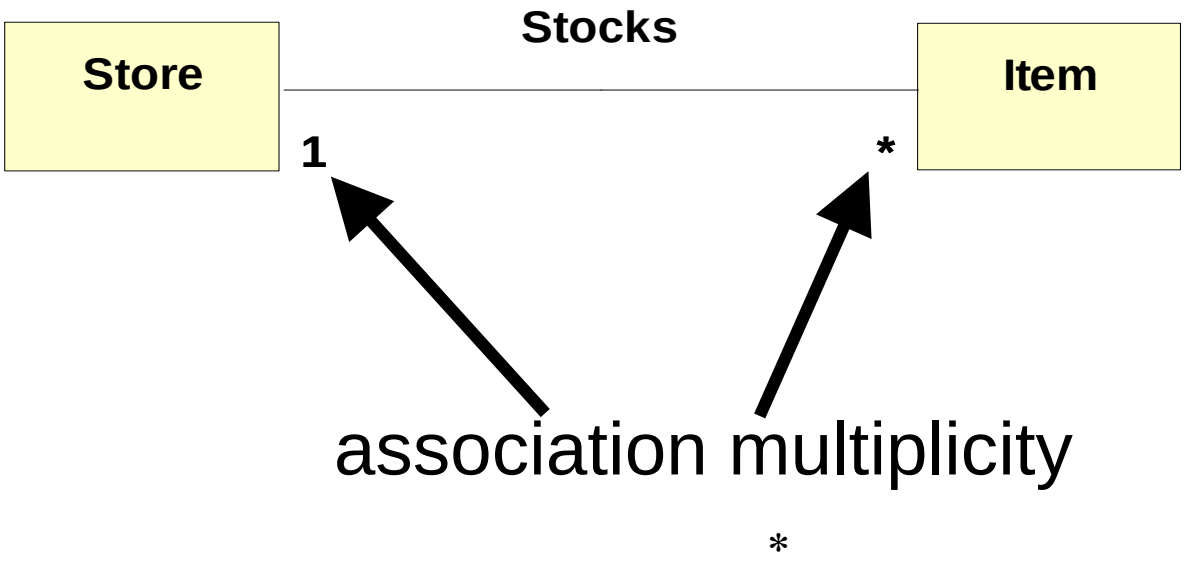
# Adding associations: Notations



# Adding associations: Multiplicity

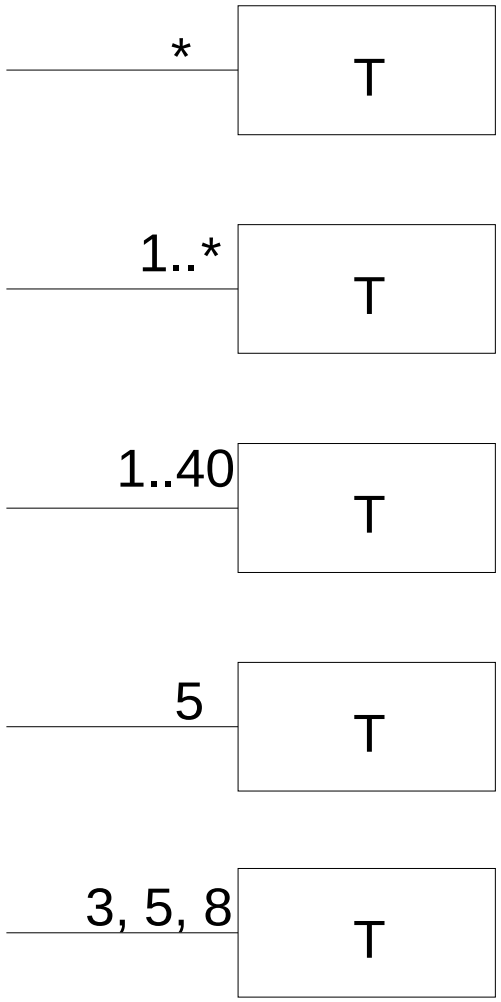


- Multiplicity defines how many instances of a class *A* can be associated with one instance of a class *B*.



# Adding associations: Multiplicity





# Adding Attributes

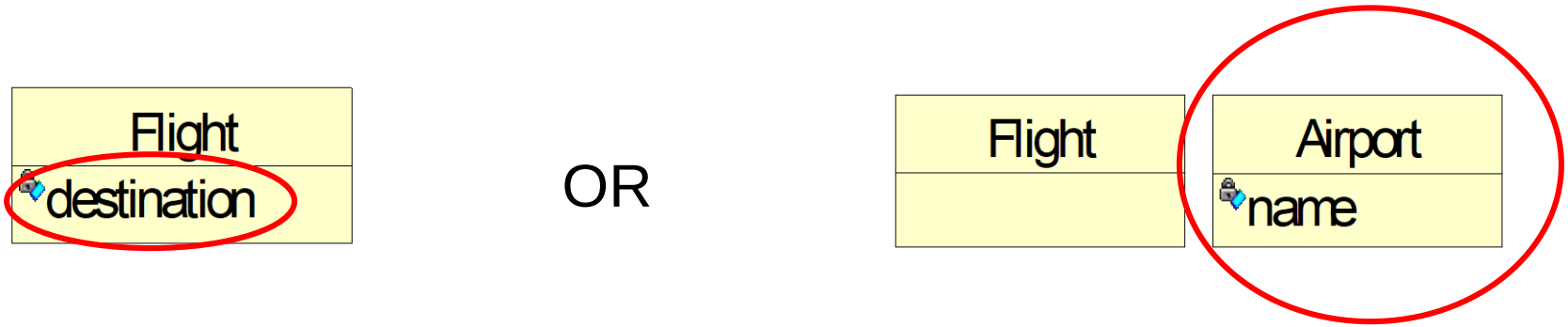


- Attributes
  - is a logical data value of an object.
  - Attributes are shown in the second compartment of the class box

# Attribute or Conceptual class?



- A common mistake when building a domain model is to represent something as an attribute when it should have been a conceptual class.




- Should destination be an attribute of flight, or a conceptual class airport?

# Attribute or Conceptual class?

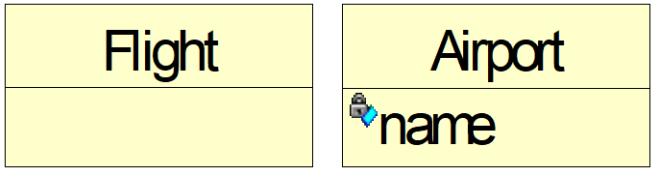


- Use the following rule of thumb:

> If we do not think of some conceptual class X as a number, date or text in the real world, then X is probably a conceptual class and not an attribute.

# Attribute or Conceptual class?

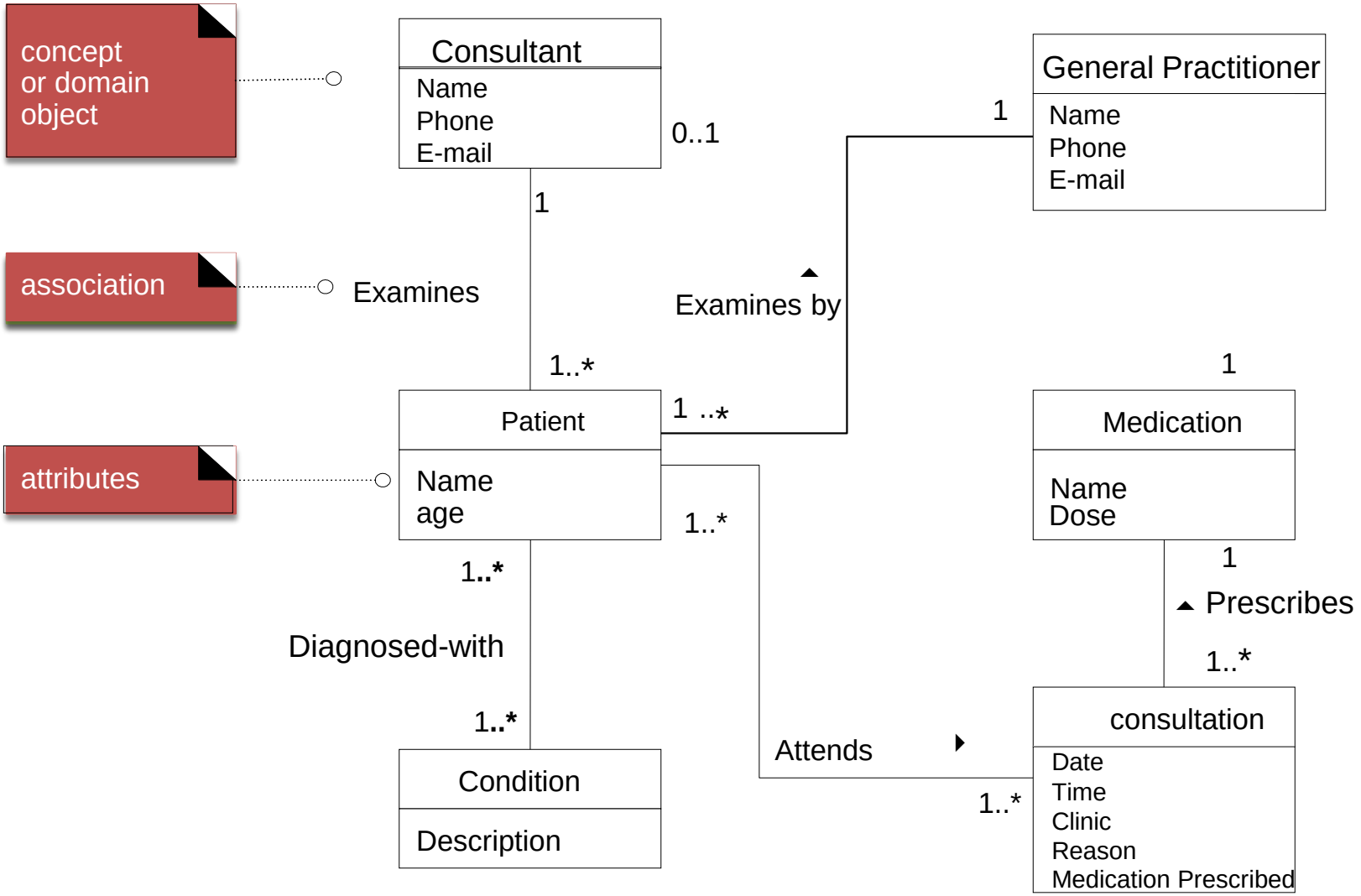





- A destination airport is a building at a specific place, it is not just a number or some text.
- Hence, it should be a conceptual class.

# Domain Model for MHC-PMS





# Assignment #4



1- Complete sections 4-3 and 4-4 in chapter 4 in your SRS

2- Send it.

Thanks