MUST

Wisdom & Virtue

MIRPUR UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF SOFTWARE ENGINEERING

Software Design & Architecture

Lecture-5

UML Communication Modeling

Saba Zafar
(Lecturer)

Date: 11-12- 2024

LECTURE CONTENTS

COMMUNICATION DIAGRAM

- In UML, a communication diagram shows the interactions between the objects or roles associated with lifelines and the messages that pass between lifelines.
- **Communication diagram** (called **collaboration diagram** in UML 1.x) is a kind of UML **interaction diagram** which shows interactions between objects and/or **parts** (represented as **lifelines**) using sequenced messages in a free-form arrangement.
-

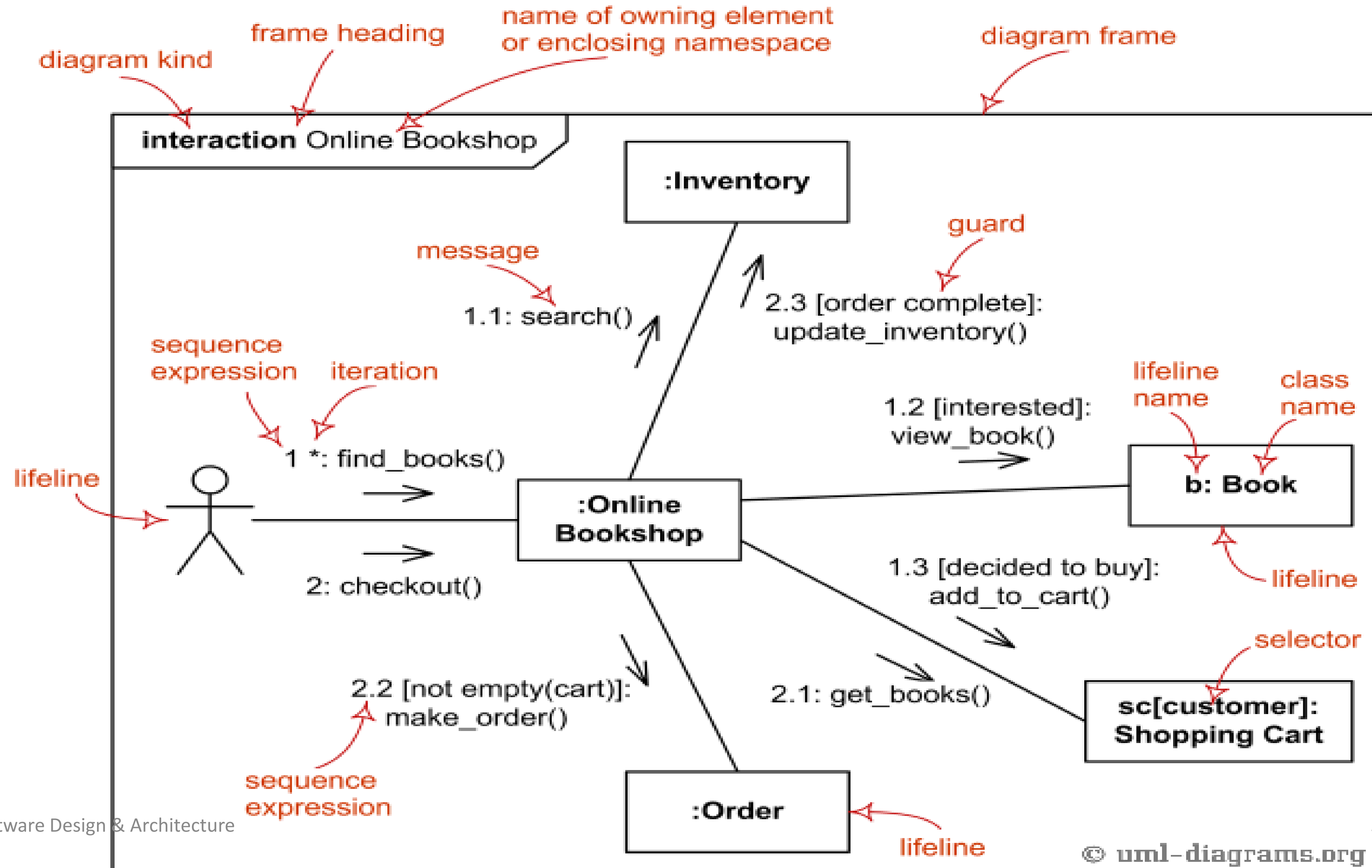


diagram kind
frame heading
name of owning element or enclosing namespace
diagram frame
interaction Online Bookshop
:Inventory
guard
message
1.1: search()
2.3 [order complete]: update_inventory()
sequence expression
iteration
lifeline name
class name
1.2 [interested]: view_book()
1 *: find_books()
lifeline
b: Book
:Online Bookshop
lifeline
2: checkout()
1.3 [decided to buy]: add_to_cart()
selector
2.2 [not empty(cart)]: make_order()
2.1: get_books()
sc[customer]: Shopping Cart
sequence expression
:Order
lifeline
© uml-diagrams.org

COMMUNICATION DIAGRAM

- The term interaction diagram is a generalization of two more specialized UML diagram types:
  a. Sequence diagrams
  b. Communication diagrams
- Sequence diagrams are the more notationally rich, but communication diagrams have their use as well, especially for wall sketching
- Sequence diagrams illustrate interactions in a kind of fence format, in which each new object is added to the right
- Communication diagrams illustrate object interactions in a graph or network format, in which objects can be placed anywhere on the diagram
- Communication Diagram is more focused on showing the collaboration of objects rather than the time sequence

PURPOSE OF COMMUNICATION DIAGRAM

- Model message passing between objects or roles that deliver the functionalities of use cases and operations
- Support the identification of:
  - Objects (hence classes)
  - Attributes (parameters of message)
  - Operations (messages) that participate in use cases

PURPOSE OF COMMUNICATION DIAGRAM

- The main purpose of a UML communication diagram is to show the messages that objects send to one another (or themselves) to make the system do what a use case specifies.
- UML communication diagrams are useful during design to visualize the proper sequence of messages between objects and actors. Mapping this out helps developers and programmers save time and energy, plus avoid making mistakes early on. These diagrams also serve as documentation for future changes, updates, and redesigns.

When to use a UML communication diagram

- UML communication diagrams are typically created before the development of a system to map out what needs to be built and how. The purpose of diagramming messages and interactions between objects is to achieve a functionality or behavior stated in a use case or function.
- More specifically, you'd want to use a UML communication diagram to depict the sequence of messages or information. They model the flow of messages (and the order in which they occur) between objects or between a user and objects.

SEQUENCE DIAGRAM VS COMMUNICATION DIAGRAM

Figure 15.1. Sequence diagram.

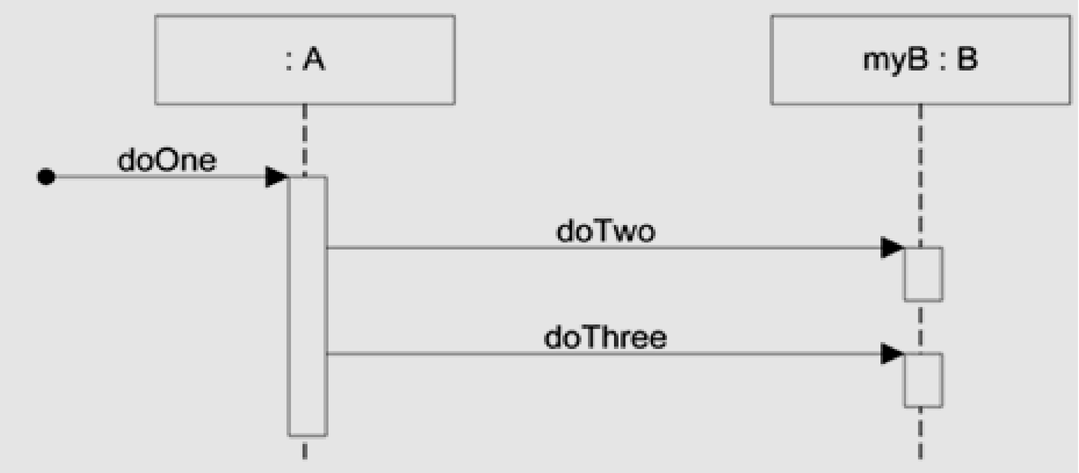
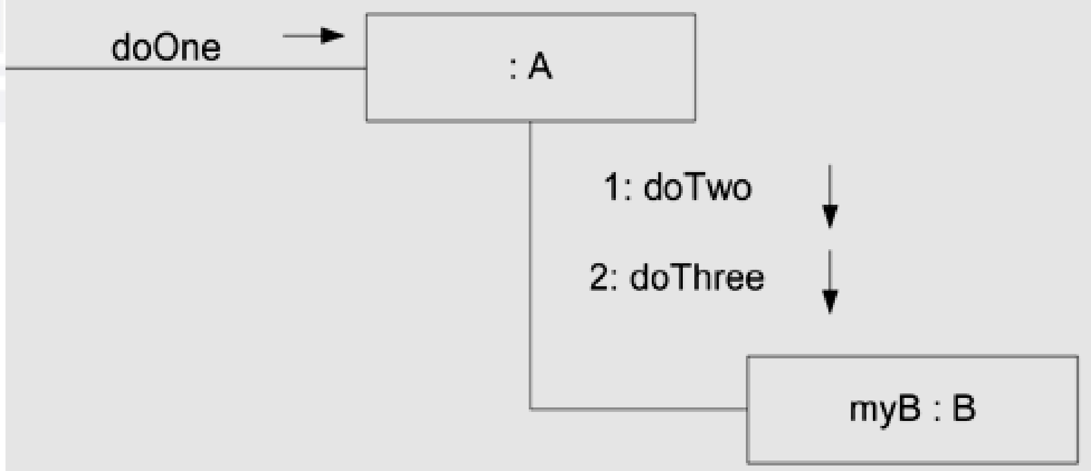
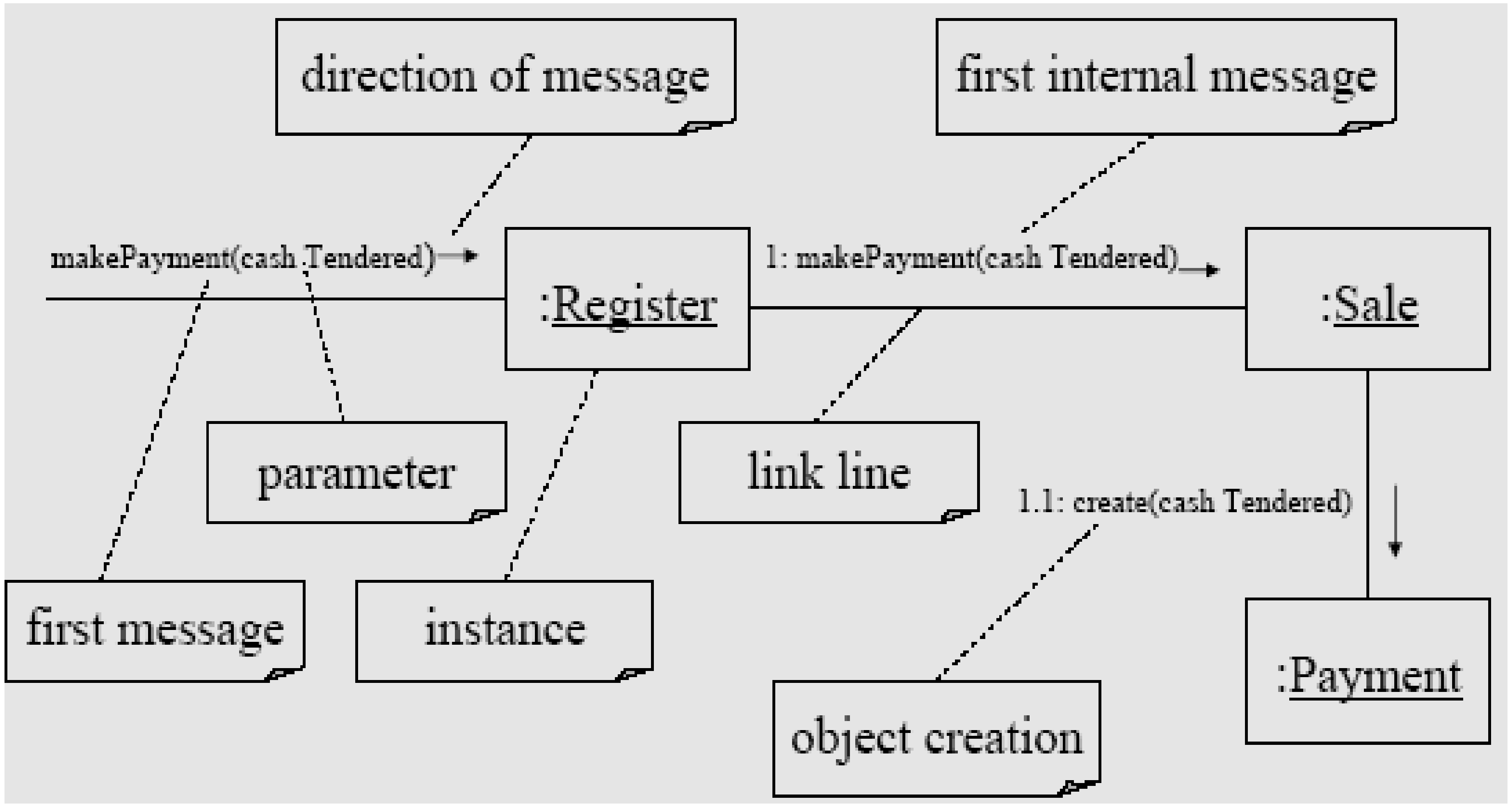


Figure 15.2. Communication diagram.



EXAMPLE: MAKE PAYMENT

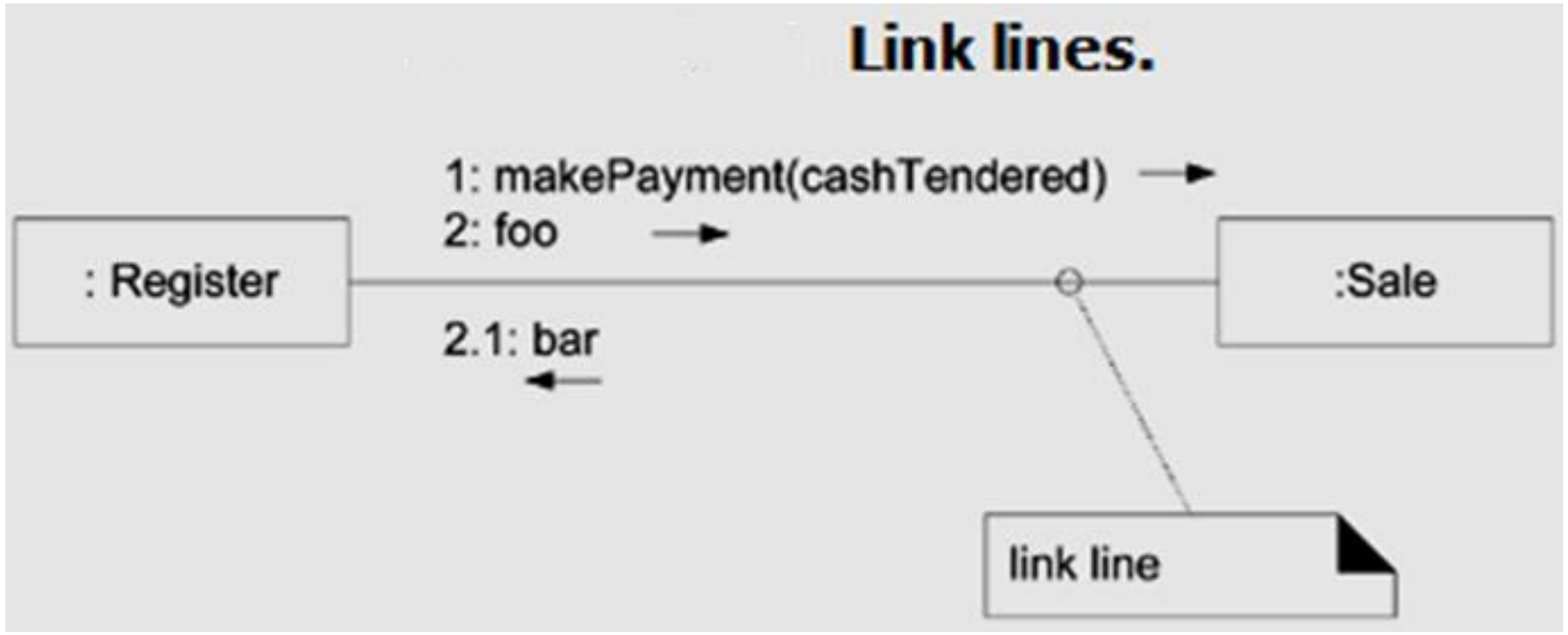


BASIC COMMUNICATION DIAGRAM NOTATION

Links

- A link is a connection path between two objects
- It indicates some form of navigation between the objects is possible
- **For example**, there is a link or path of navigation from a Register to a Sale, along which messages **makePayment** may flow
- Note that multiple messages and messages both ways, flow along the same single link
- There isn't one link line per message; all messages flow on the same line, which is like a road allowing two-way traffic

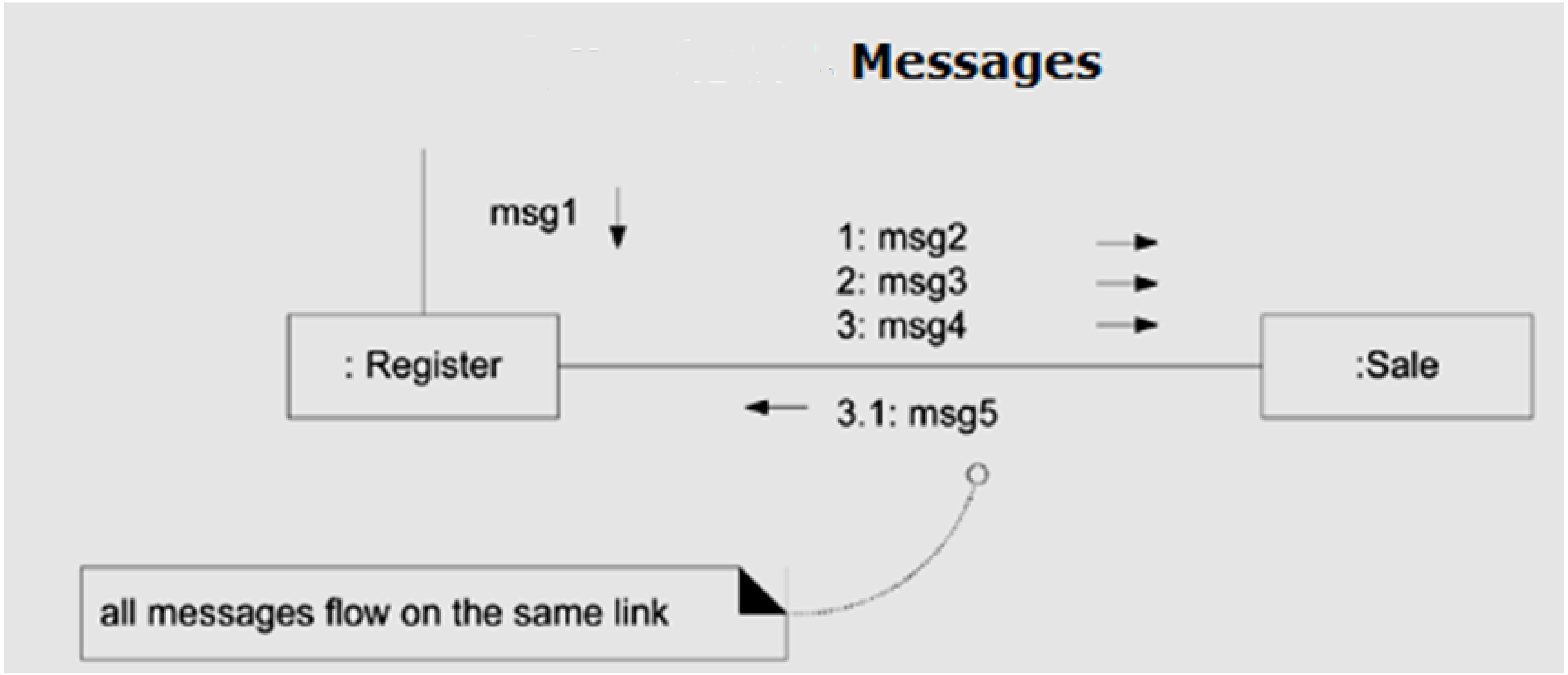
LINK


Link lines.
1: makePayment(cashTendered)
2: foo
: Register
:Sale
2.1: bar
link line

MESSAGES

- Each message between objects is represented with:
  a. A message expression
  b. Small arrow indicating the direction of the message
- Many messages may flow along this link
- A sequence number is added to show the sequential order of messages in the current thread of control
- Don't number the starting message as it's legal to do so, but simplifies the overall numbering if you don‘t

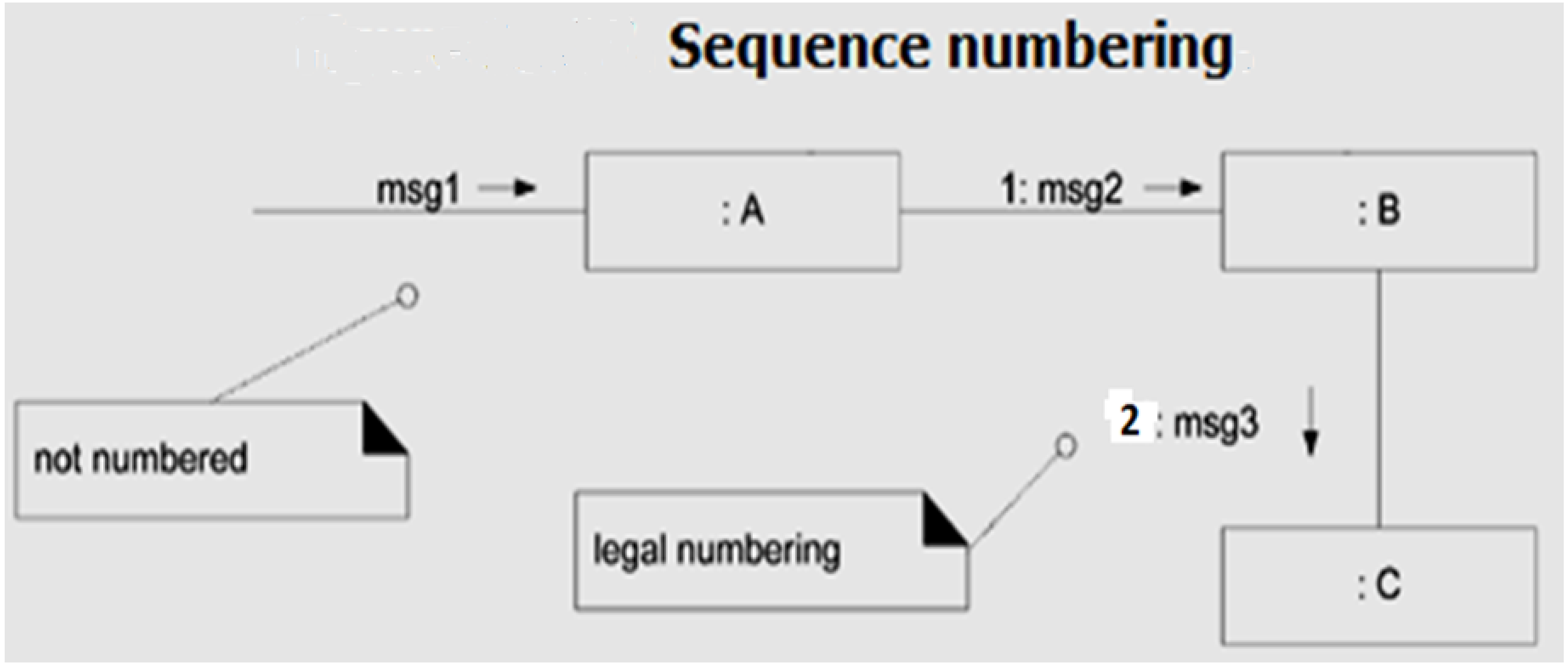
MESSAGES


Messages
msg1
1: msg2
2: msg3
3: msg4
: Register
:Sale
3.1: msg5
all messages flow on the same link

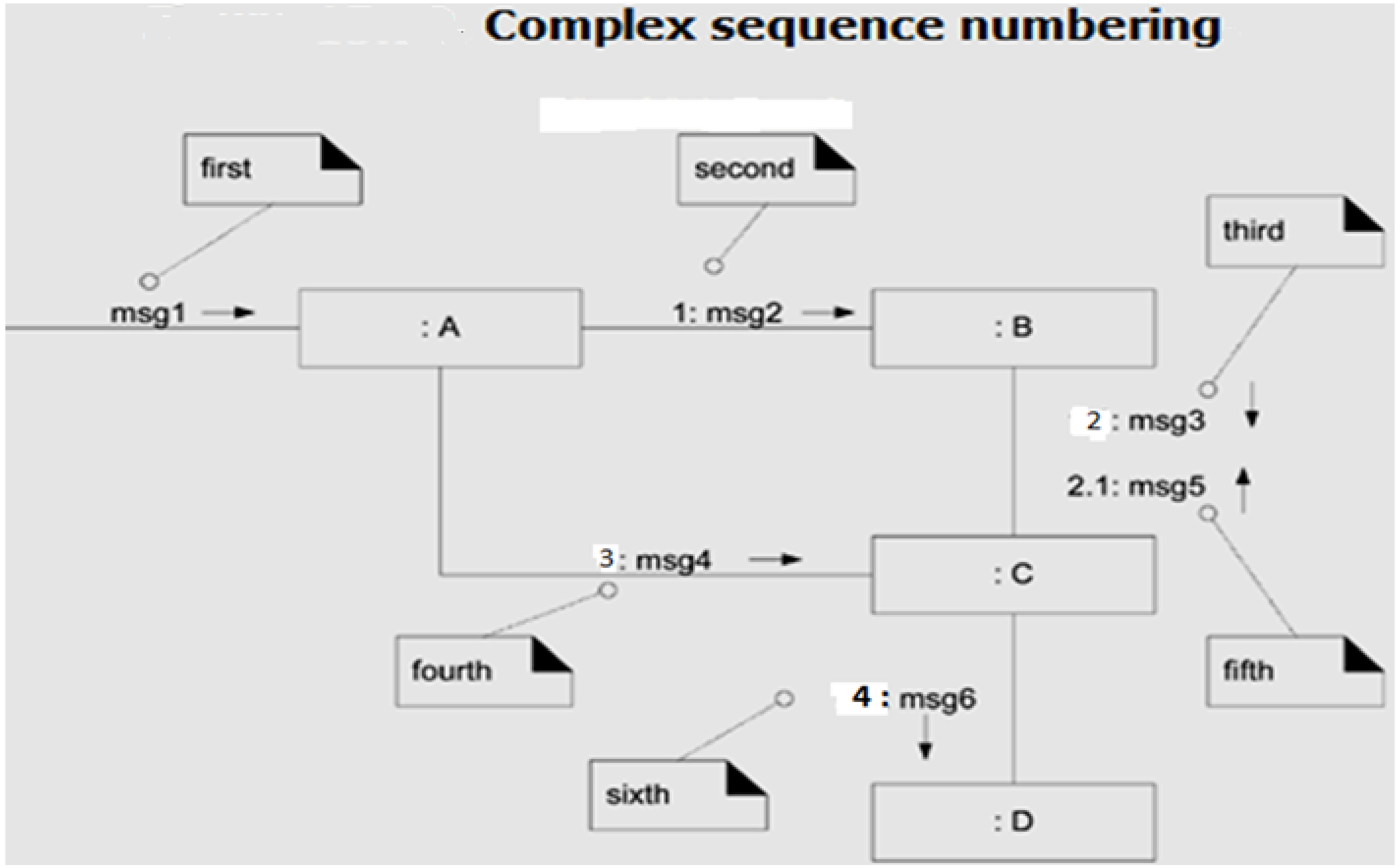
MESSAGE NUMBER SEQUENCING

- The order of messages is illustrated with sequence numbers
- The numbering scheme is:
  - The first message is not numbered. Thus, **msg1** is unnumbered
    - Actually, a starting number is legal, but it makes all subsequent numbering more awkward, creating another level of number-nesting deeper

MESSAGE NUMBER SEQUENCING (SIMPLE)



MESSAGE NUMBER SEQUENCING(COMPLEX)



STRENGTHS AND WEAKNESSES OF SEQUENCE DIAGRAMS

- Sequence diagrams have some advantages over communication diagrams
- Perhaps first and foremost, the UML specification is more sequence diagram centric, more thought and effort has been put into the notation
- Thus, tool support is better and more notation options are available
- Also, it is easier to see the call-flow sequence with sequence diagrams simply read top to bottom
- Hence, sequence diagrams are excellent for documentation or to easily read a reverse-engineered sequence, generated from source code with a UML tool

STRENGTHS AND WEAKNESSES OF COMMUNICATION DIAGRAMS

- Communication diagrams have advantages when applying "UML as sketch" to draw on walls because they are much more space-efficient
  - This is because the boxes can be easily placed or erased any where horizontal or vertical
- Modifying wall sketches is easier with communication diagrams it is simple to erase a box at one location, draw a new one elsewhere, and sketch a line to it
- In contrast, new objects in a sequence diagrams must always be added to the right edge
  - So it quickly consumes and exhausts right-edge space on a page
  - Free space in the vertical dimension is not efficiently used
- Developers doing sequence diagrams on walls rapidly feel the drawing pain when contrasted with communication diagrams

SEQUENCE vs COMMUNICATION DIAGRAM

Type	Strengths	Weaknesses
Sequence	• Clearly shows sequence or time ordering of messages	• Forced to extend to the right when adding new objects • Consumes horizontal space
Communication	• Space economical • Flexibility to add new objects in two dimensions	• More difficult to see sequence of messages

INTRODUCTION

- Object diagrams represent an instance of a class diagram
- The basic concepts are similar for class diagrams and object diagrams
- Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment
- Object diagrams are used to concentrate on a set of objects and their relationships as an instance

Object diagrams

- Object is an instance of a class in a particular moment in runtime that can have its own state and data values. Likewise a static **UML** object diagram is an instance of a **class diagram**; it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships which may be considered a special case of a class diagram or a **communication diagram**.

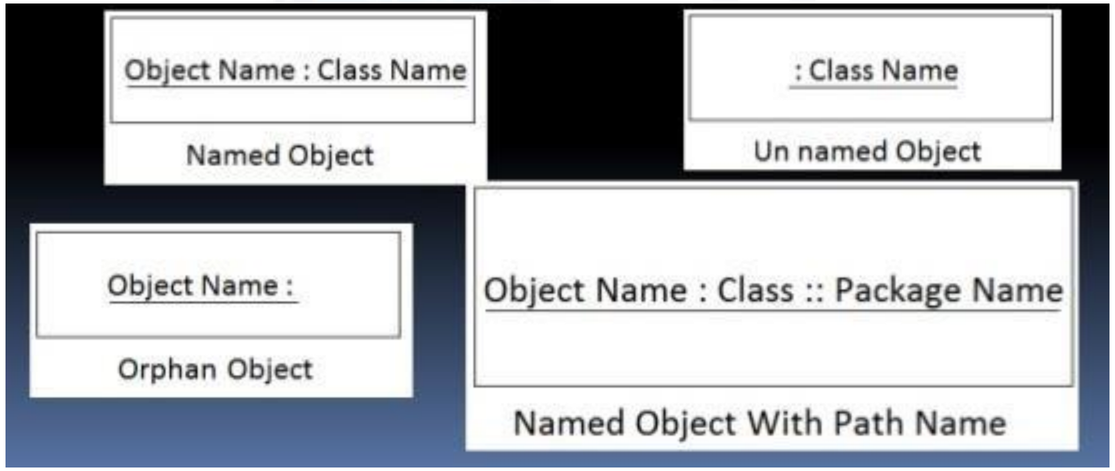
PURPOSE OF OBJECT DIAGRAM

- It shows object relationships in a system
- Understand object behavior and their relationships
- Shows static view of a system
- It can be used to perform forward and reverse engineering on systems
- It can be used to explore the relations of an object and can be used to analyze other connecting objects

TERMS AND CONCEPTS OF OBJECT DIAGRAM

1. Object

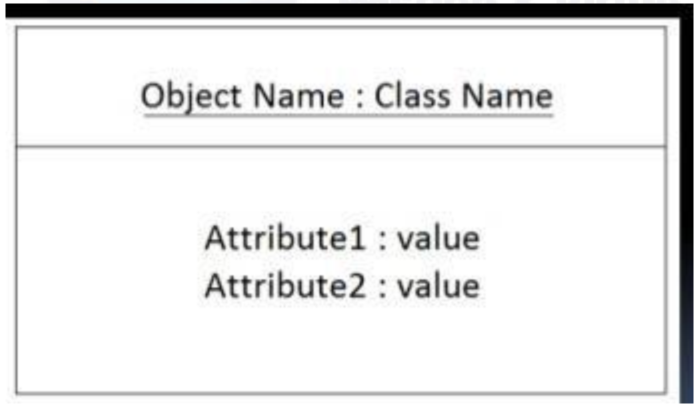
- It is instance of a class and each object is represented by a rectangle which contains name of object and its class name underlined and separated by a colon.



TERMS AND CONCEPTS OF OBJECT DIAGRAM

2. Object with Attributes

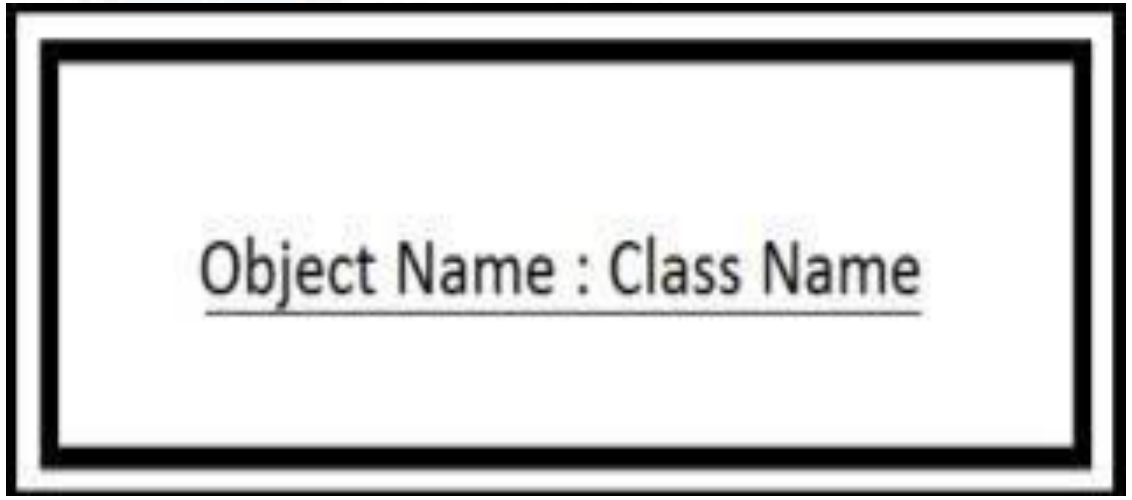
- As with classes we can list object attributes in separate compartment
- Each object attributes must have values assigned to them.



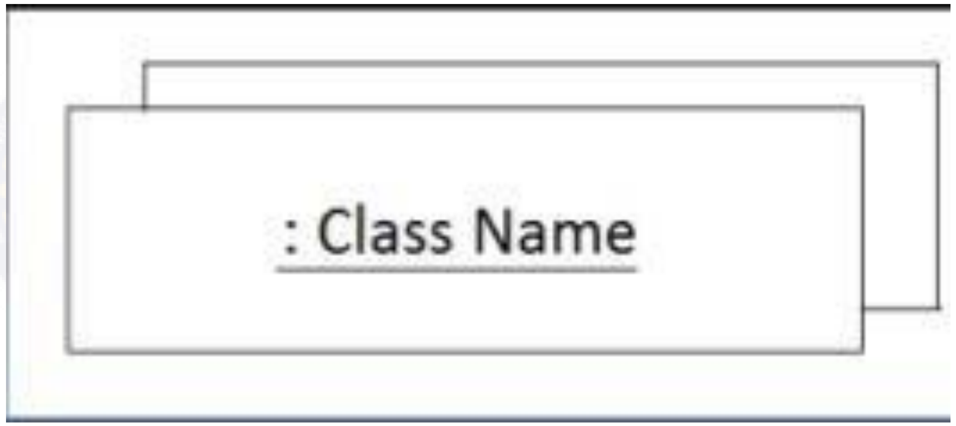
TERMS AND CONCEPTS OF OBJECT DIAGRAM

3. Active Objects

- Objects that have their own processes that control their activities are called as active objects
- They are represented by a symbol of object with thick border lines



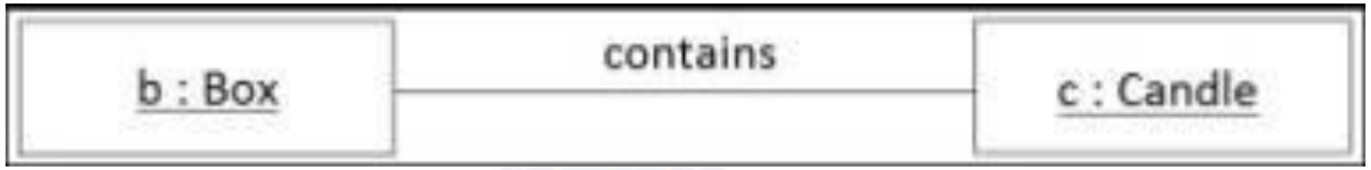
TERMS AND CONCEPTS OF OBJECT DIAGRAM



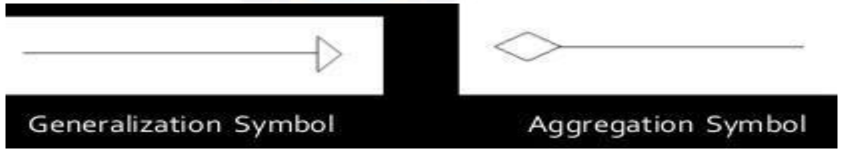
4. Multiple Objects : It is represented by a symbol as object if attributes of objects are not important.

TERMS AND CONCEPTS OF OBJECT DIAGRAM

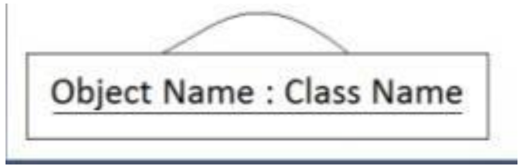
5. Links: These are the relationship between objects. Links are instances of association. It is represented by solid line.



Relationship between objects can also be generalization or aggregation.



Objects that fulfill more than role can be self linked.

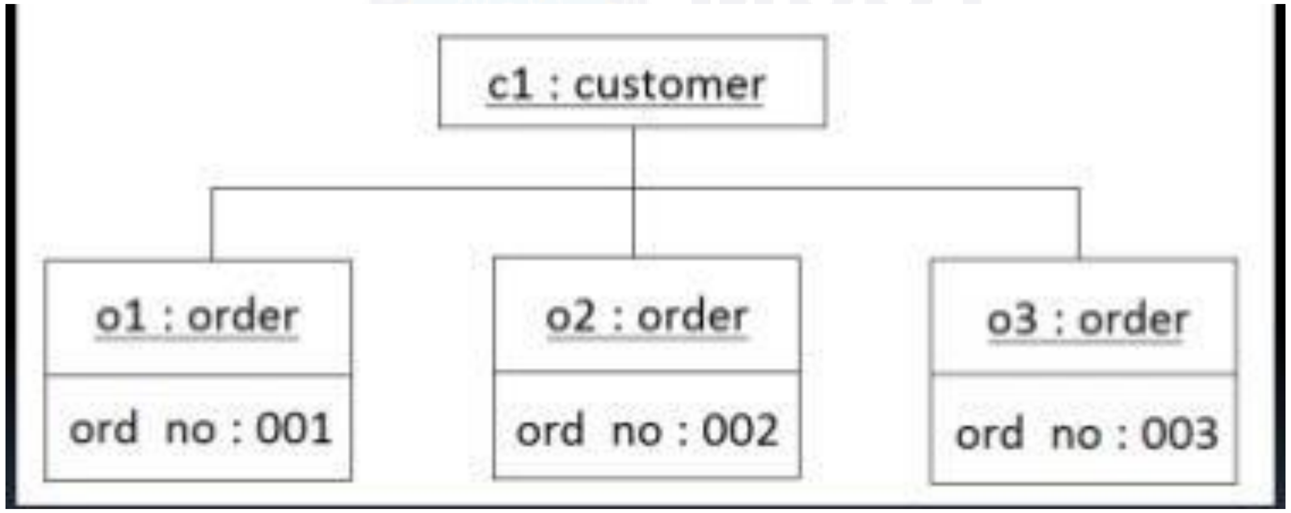


OBJECT INSTANCES

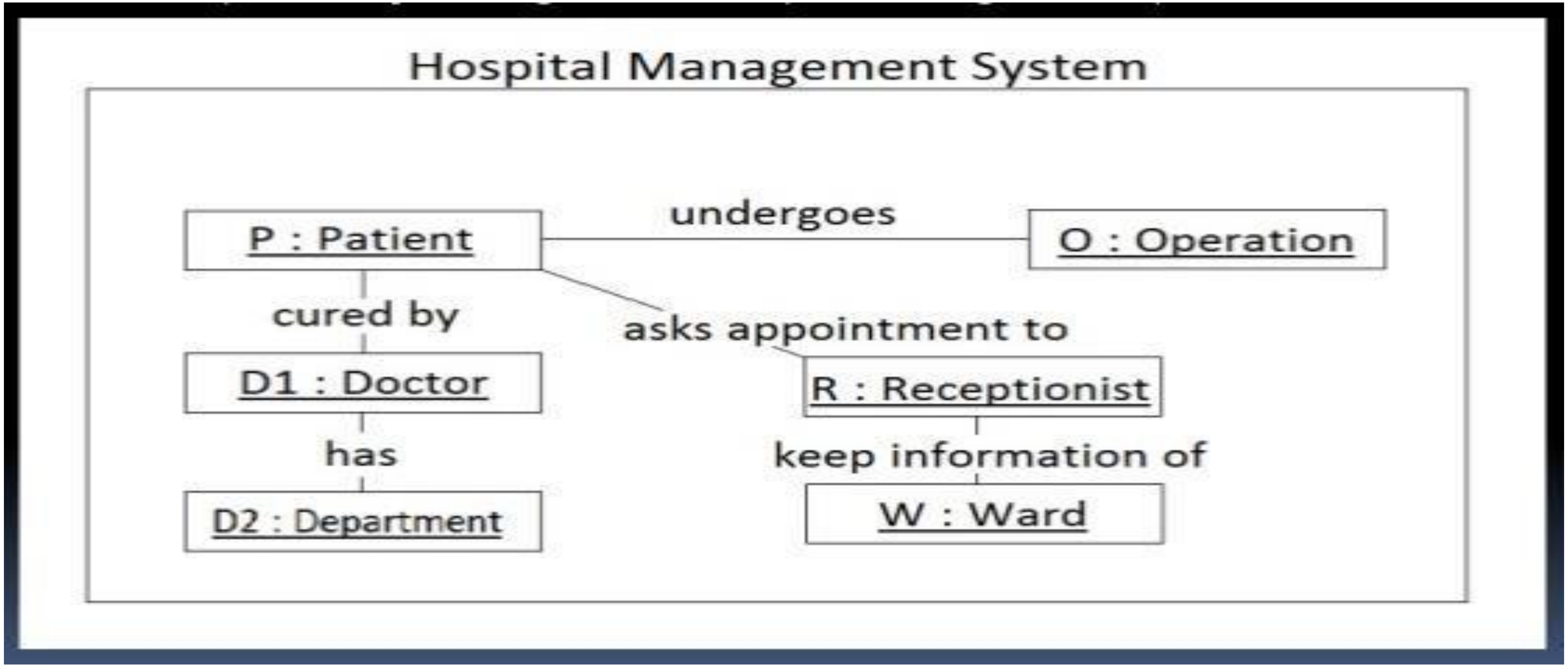
Object instance is also called as class object that it is nothing but instance of a class. Objects are extracted as instance usually on object diagrams.

Example:

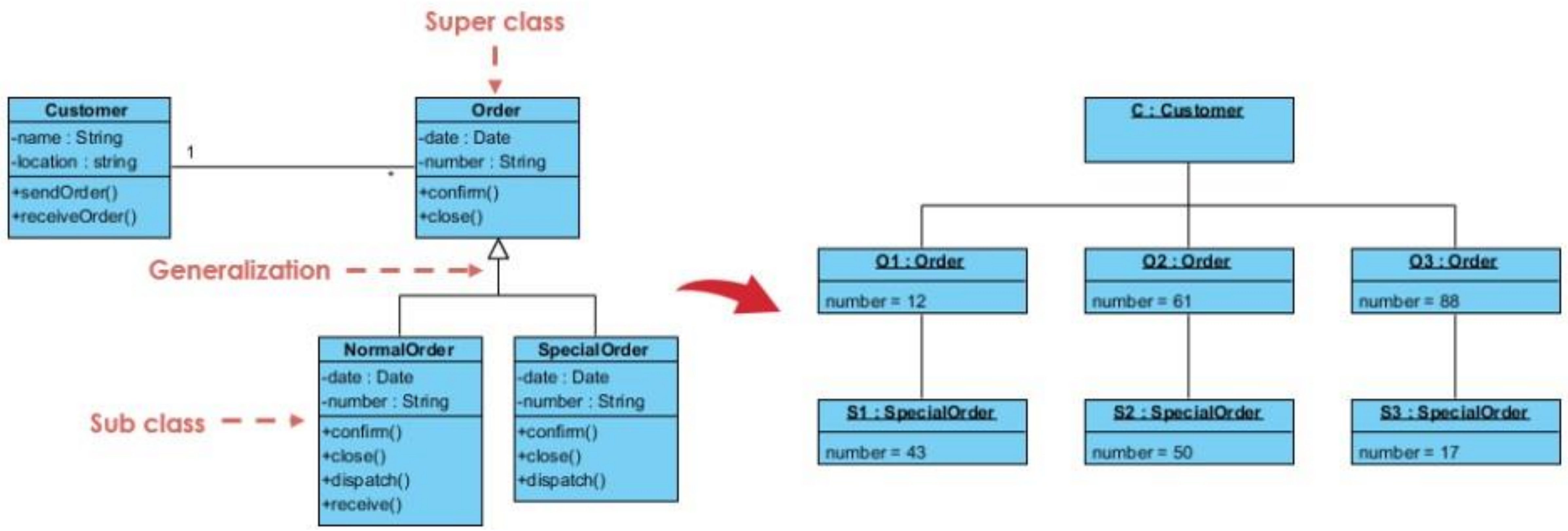
Following diagram shows instance of a class order and of class customer.



EXAMPLE OF OBJECT DIAGRAM OF HOSPITAL MANAGEMENT



EXAMPLE: CLASS TO OBJECT DIAGRAM (Order System)



THANKS