



# SS ZG514

## Object Oriented Analysis and Design



**BITS Pilani**

Pilani | Dubai | Goa | Hyderabad

Dr. Ritu Arora  
[rituarora@pilani.bits-pilani.ac.in](mailto:rituarora@pilani.bits-pilani.ac.in)

Content s of these slides are adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition



# Interaction Diagrams

# Interaction Diagram

---

- Shows interaction among objects within a system
- Dynamic or behavioral diagrams
- Scenario specific diagram

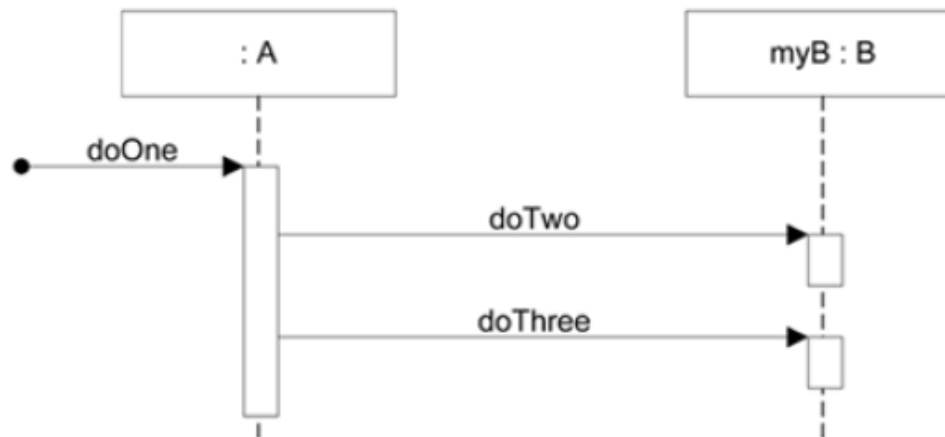
Two types:

- Sequence Diagram
  - Communication Diagram
- 
- Both differ in format, layout and notations.
  - Draw only one type to depict the interaction among objects.

# Sequence Diagram



- Illustrate interactions in a kind of fence format.
- Shows interaction between the objects within the System.
- Messages are layered in order to depict the sequence of messages. Hence no need to number the messages.
- Execution Specification bar is used to represent focus of control. Uncommon in UML sketching.

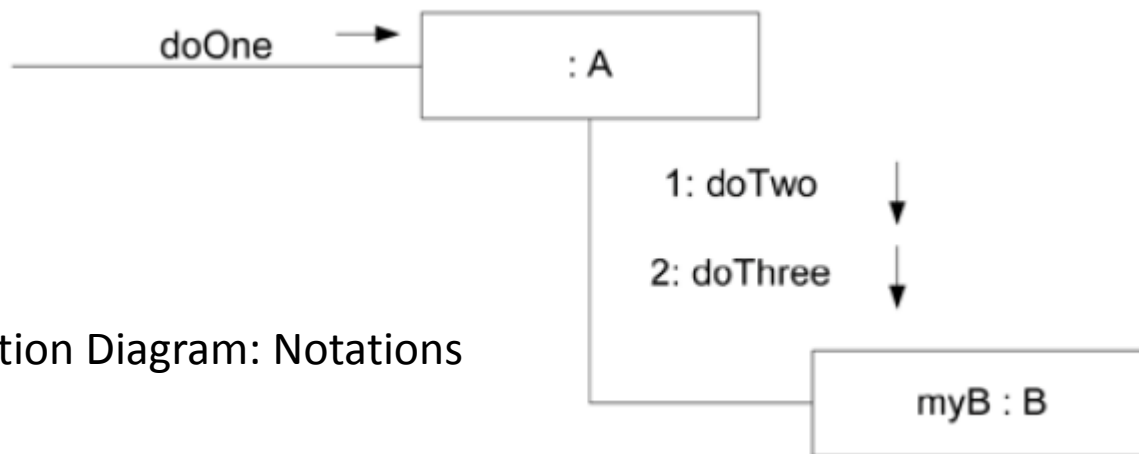


Sequence Diagram: Notations

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

# Communication Diagram

- Illustrates object interaction in a graph or network format
- Do not represent the external actors
- Messages to be numbered sequentially; drawn as solid lines
- Separate arrows are drawn to represent the direction of messages
- Do not number the first message



Communication Diagram: Notations

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

# Communication or Sequence Diagram



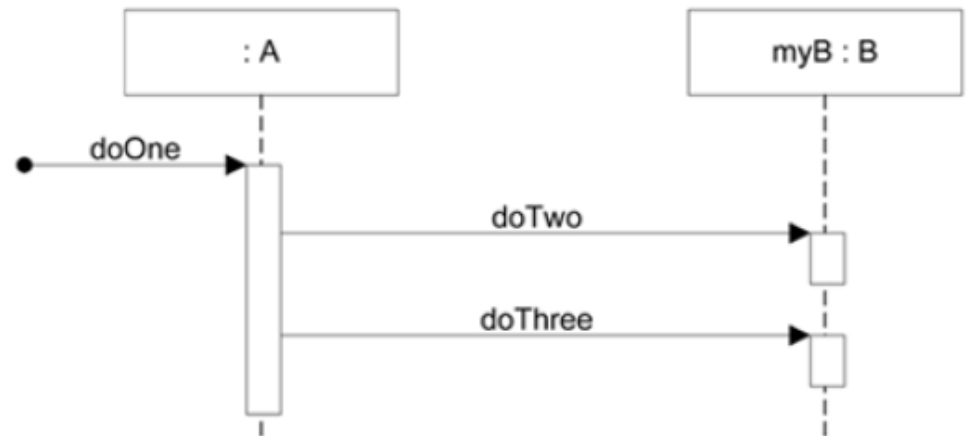
- If the numbers of messages is too large -> go for sequence diagram.
- If the number of objects is very large - > go for communication diagram



# Sequence Diagrams

# Representation in code

```
public class A
{
    private B myB = new B();
    public void doOne()
    {
        myB.doTwo();
        myB.doThree();
    }
    // ...
}
```



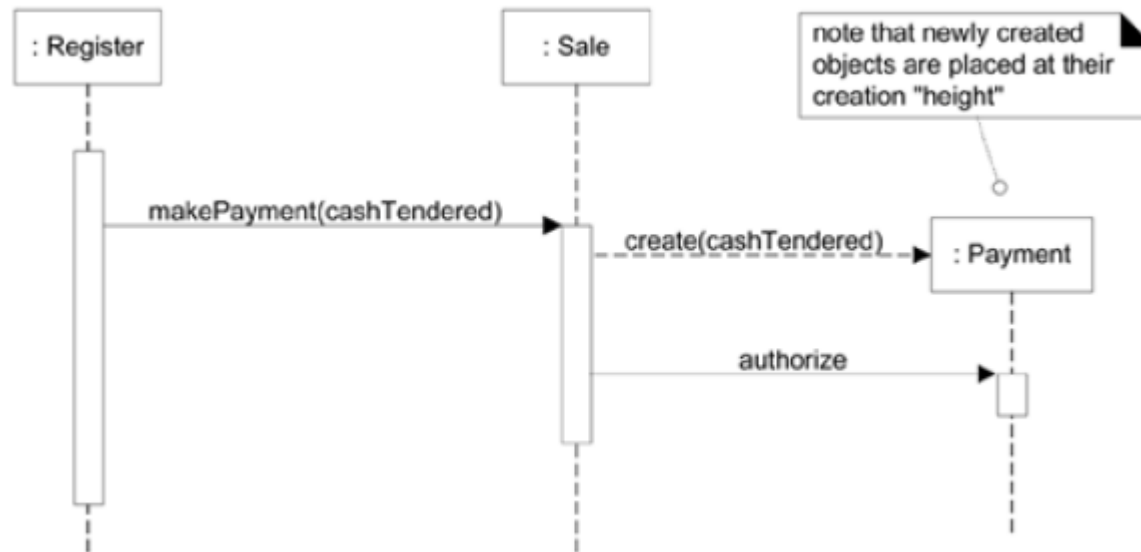
Sequence Diagram



# Creation of instances: Sequence Diagram



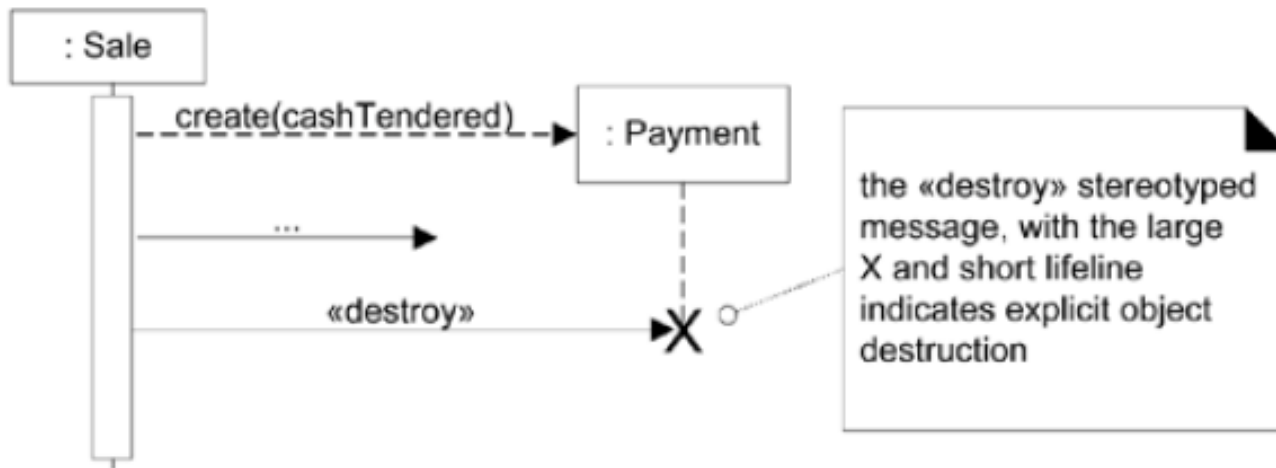
- A newly created object is placed at their creation height.



Sequence Diagram: Notations

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

# Object Destruction: Sequence Diagram



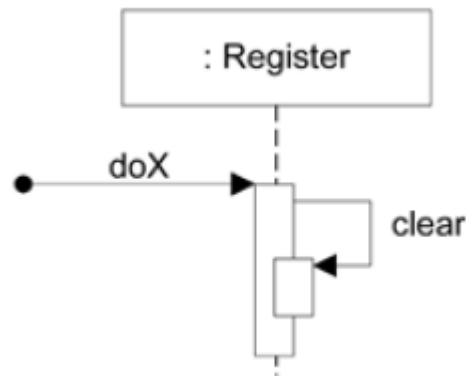
## Sequence Diagram: Notations

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

# Messages to 'self' or 'this': Sequence Diagram



- A message can be sent from an object to itself.
- This is illustrated by a link to itself, with messages flowing along the link.



Sequence Diagram: Notations

# Diagram frames: Sequence Diagram



- Used to support conditional and looping constructs
- Frames are regions or fragments of the diagram
- Consist of a operator or label (such as loop) and a guard (conditional clause)
- The guard clause should be placed over the lifeline to which it belongs

# Frame Operators



Frame Operator	Meaning
alt	Alternative fragment for mutual exclusion conditional logic expressed in the guards.
loop	Loop fragment while guard is true. Can also write <i>loop (n)</i> to indicate looping n times.
opt	Optional fragment that executes if guard is true.

# LOOP frame

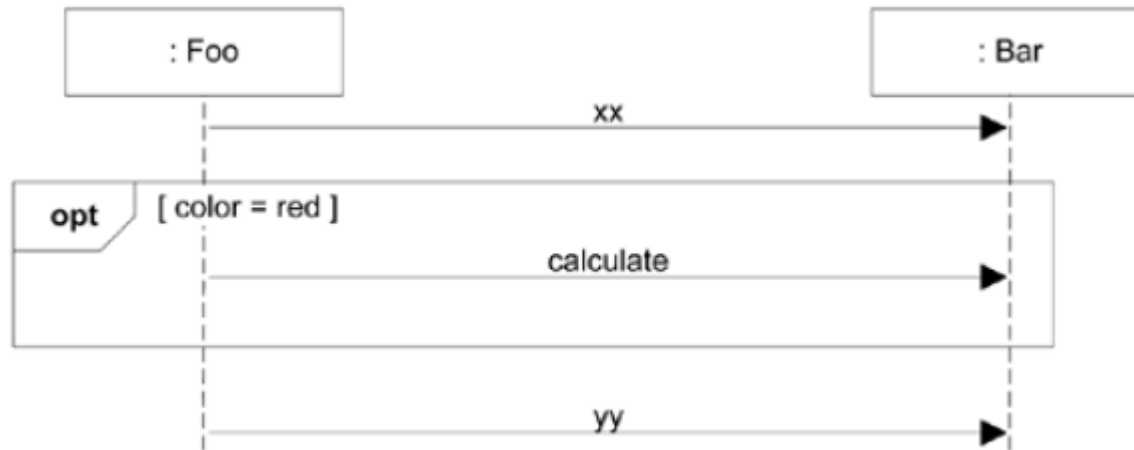


Sequence Diagram: Notations for loop interaction frame

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

# Conditional Messages

- OPT frame is placed around one or messages.
- Guard is placed over the related lifeline.

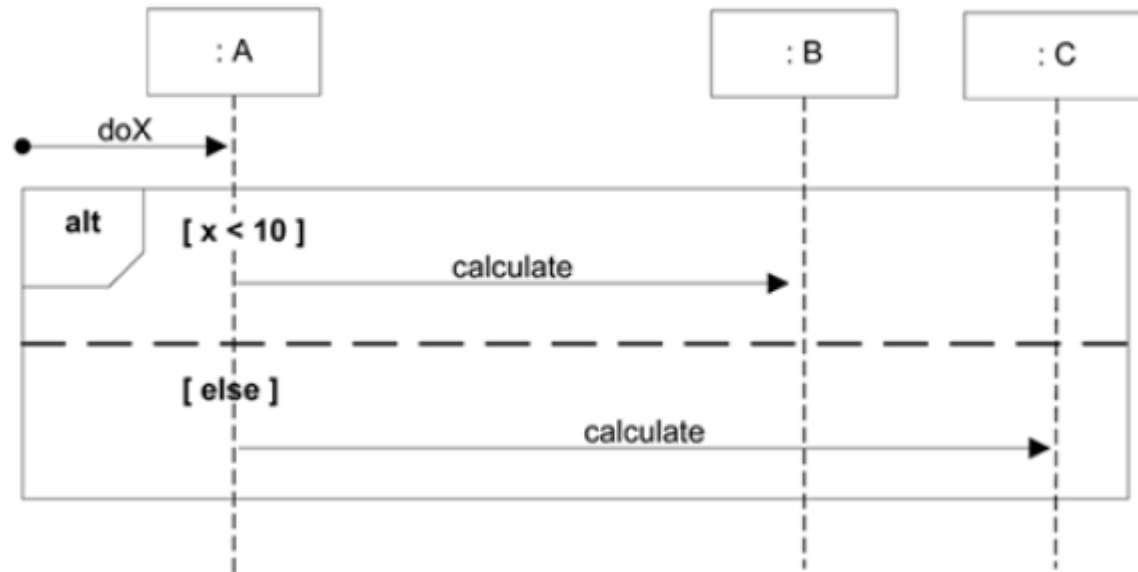


Sequence Diagram: Notations for opt interaction frame

# Mutually exclusive Conditional Messages



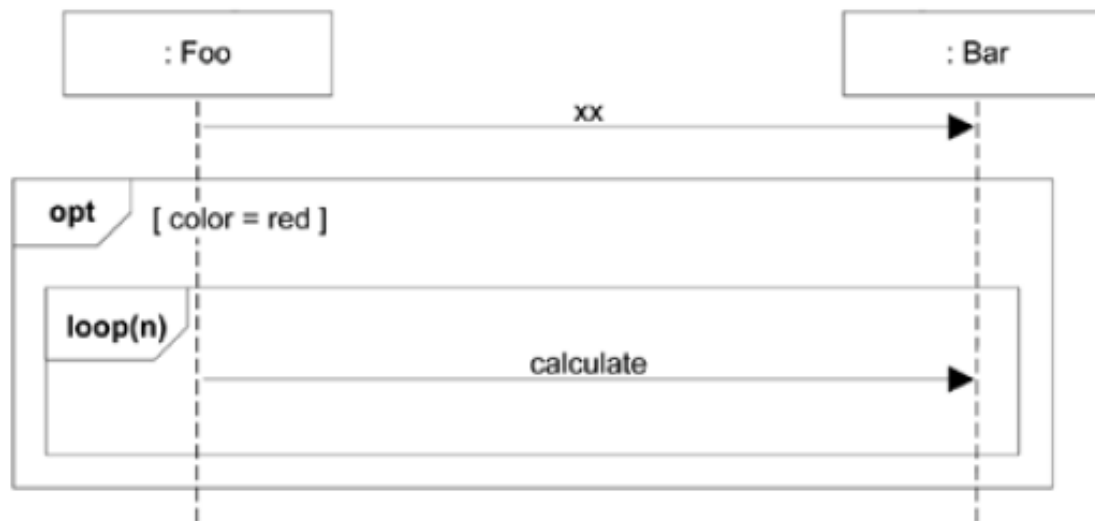
- ALT frame is placed around mutually exclusive alternatives



Sequence Diagram: Notations for alt interaction frame



# Nesting of frames



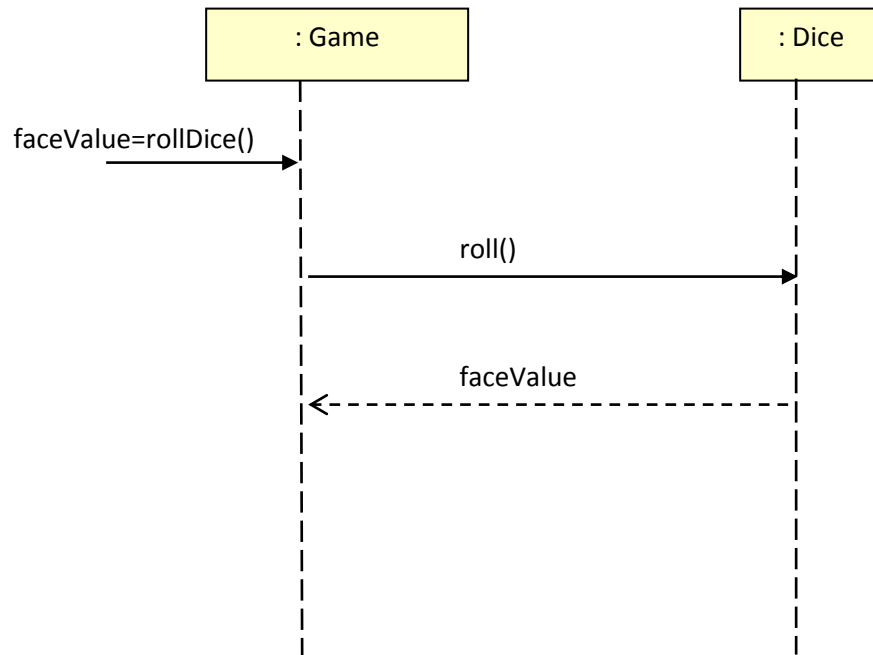
Sequence Diagram: Notations for depicting nesting of frames

# Exercise: Sequence Diagram



For the Snakes and Ladders case study, draw the sequence diagram for roll the dice scenario.

# Solution:

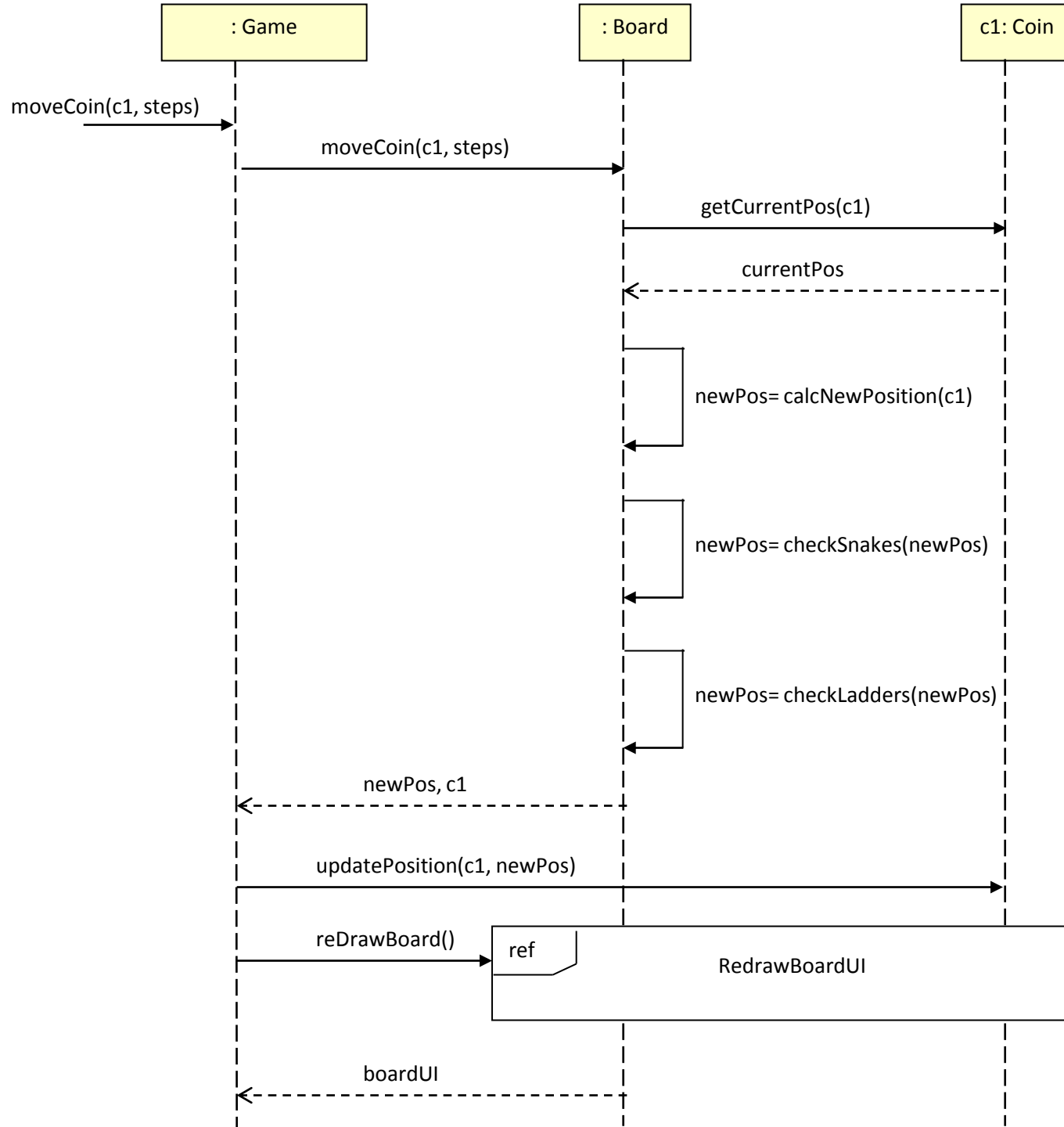


# Exercise: Sequence Diagram

---



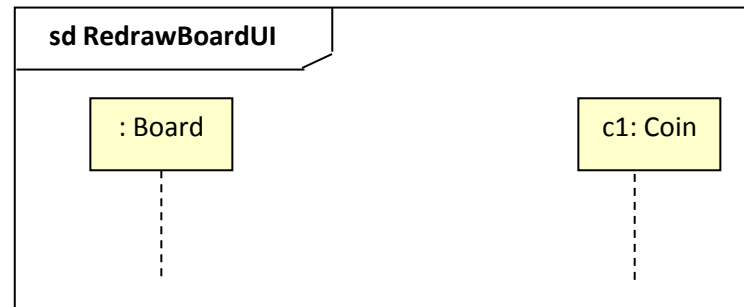
For the Snakes and Ladders case study, draw the sequence diagram for moving a coin on the board.



# Relating Sequence Diagrams



- An interaction occurrence (also called interaction use) is a reference to an interaction within another interaction.
- Useful for simplifying diagrams or for reusability of diagrams.
- Created using two related frames:
  - a frame around an entire sequence diagram, labeled with the tag `sd` and a name
  - a frame tagged `re`, called a reference, that refers to another named sequence diagram





# Communication Diagrams

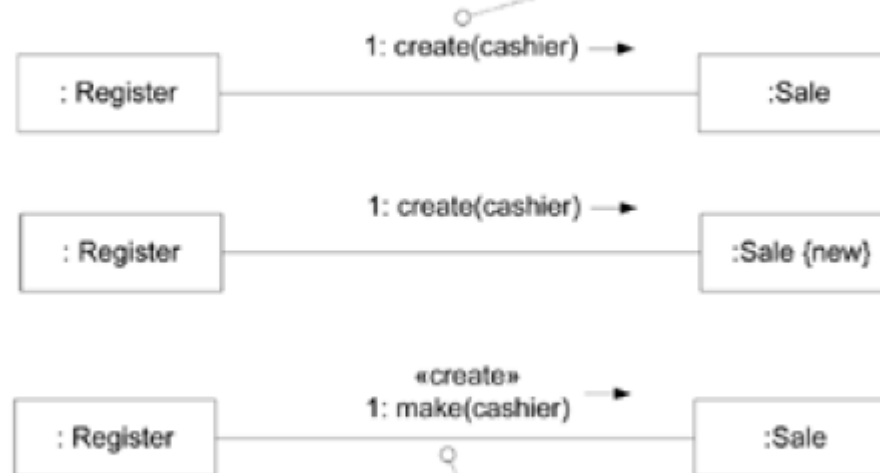
# Creation of instances: Communication Diagram



- An object might be responsible for creation of another object
- A create message with associated parameters is sent

Three ways to show creation in a communication diagram

create message, with optional initializing parameters. This will normally be interpreted as a constructor call.



if an unobvious creation message name is used, the message may be stereotyped for clarity

## Communication Diagram: Notations

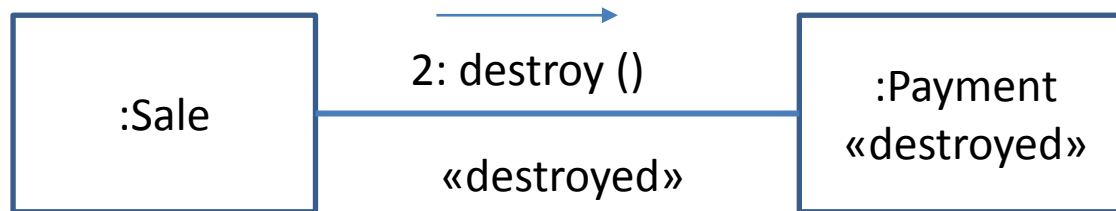
Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition



# Object Destruction: Communication Diagram



- A destroy message is sent to end the life of an object
- Use the «destroy» stereotype tag to emphasize destruction or end of life of the object



Destroy messages in Communication Diagram

# Messages to 'self' or 'this': Communication Diagram



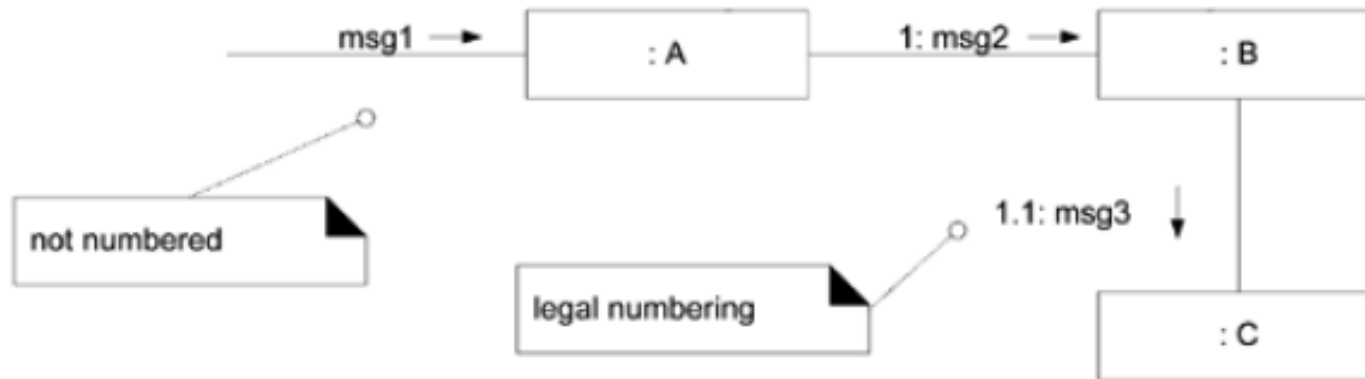
- A message can be sent from an object to itself.
- This is illustrated by a link to itself, with messages flowing along the link.



Communication Diagram: Notations

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

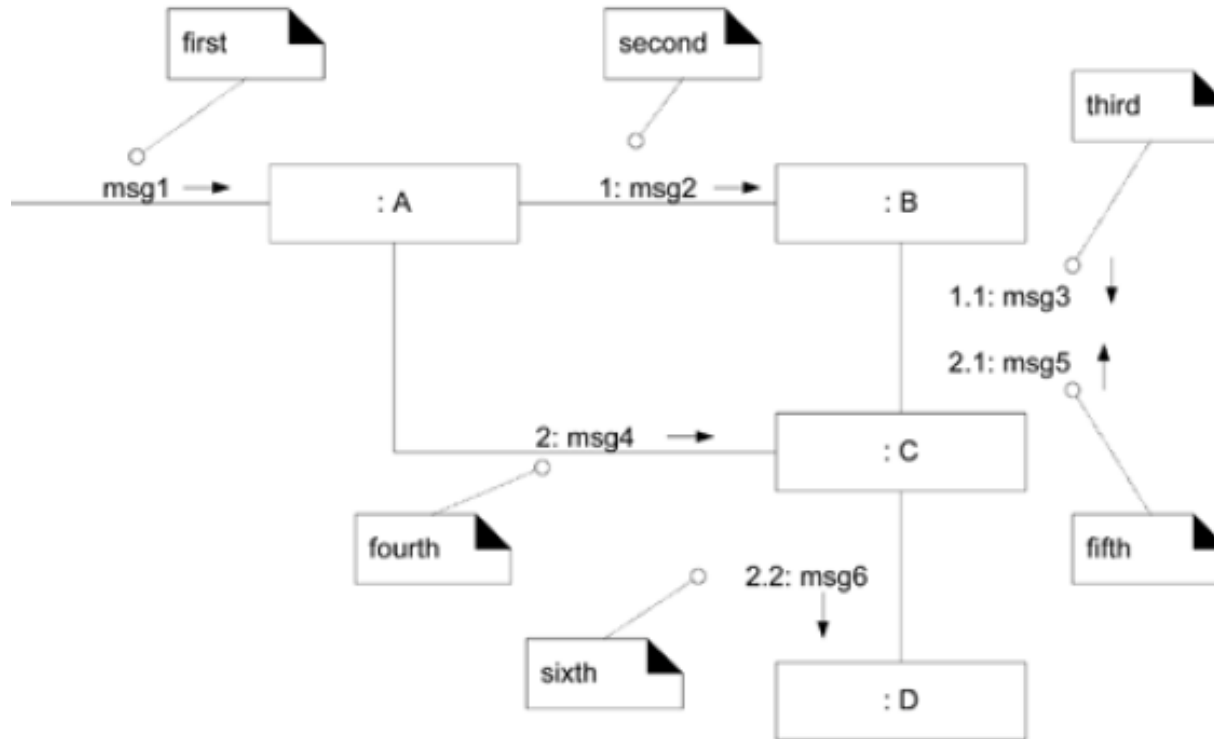
# Message numbering: Communication Diagram



Simple numbering of messages in Communication Diagram

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

# Message numbering: Communication Diagram



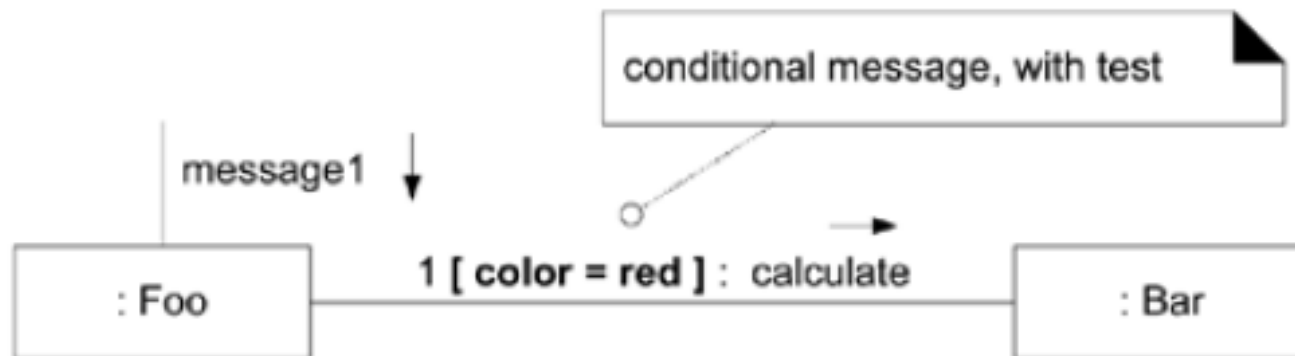
Indexed numbering of messages in Communication Diagram

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

# Conditional Messages: Communication Diagrams



- Place the conditional message within square brackets after the sequence number.
- The message is only sent if the clause evaluates to true.

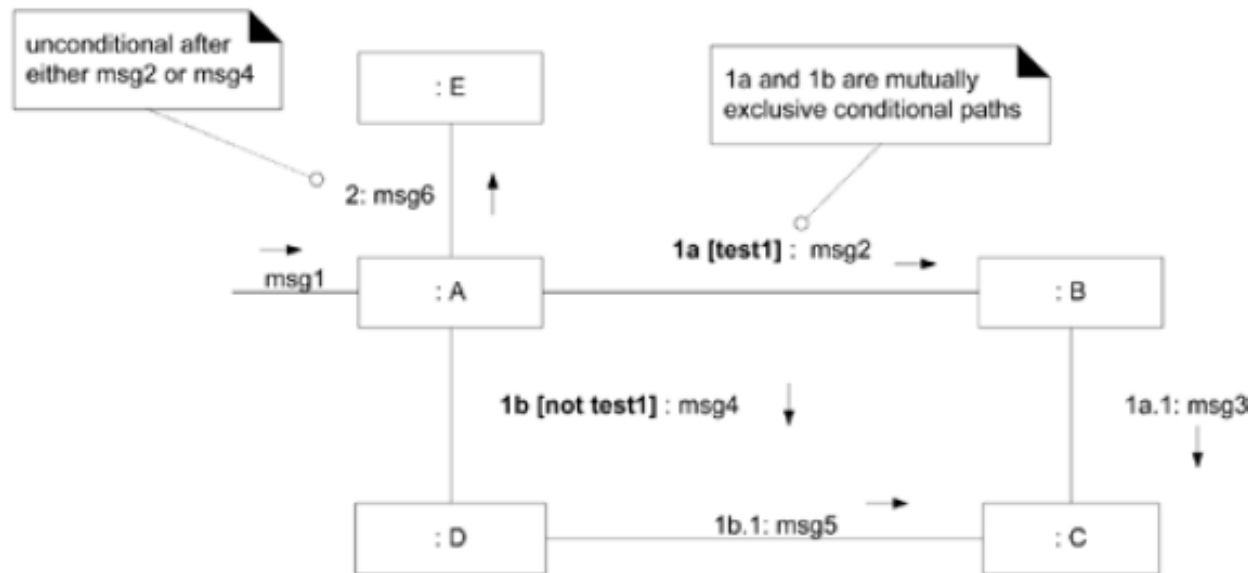


Conditional messages in Communication Diagram

# Mutually exclusive conditional paths



- Modify the sequence number for mutually exclusive conditions

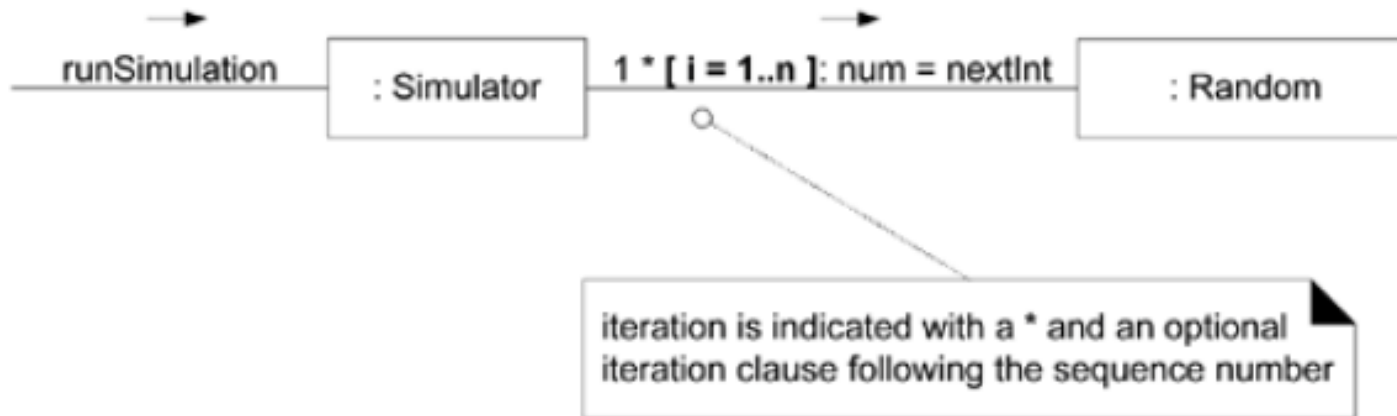


Mutually exclusive conditional paths in Communication Diagram

# Iterations: Communication Diagrams



- Iteration is represented using a \*
- An optional iteration clause can be placed following the sequence number

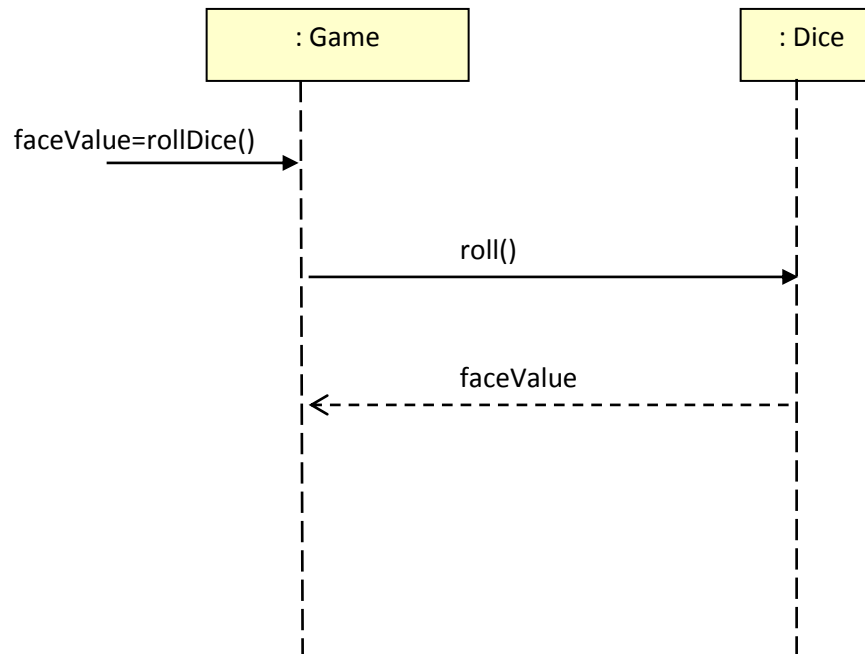


Iterations in Communication Diagram

# Exercise: Communication Diagram

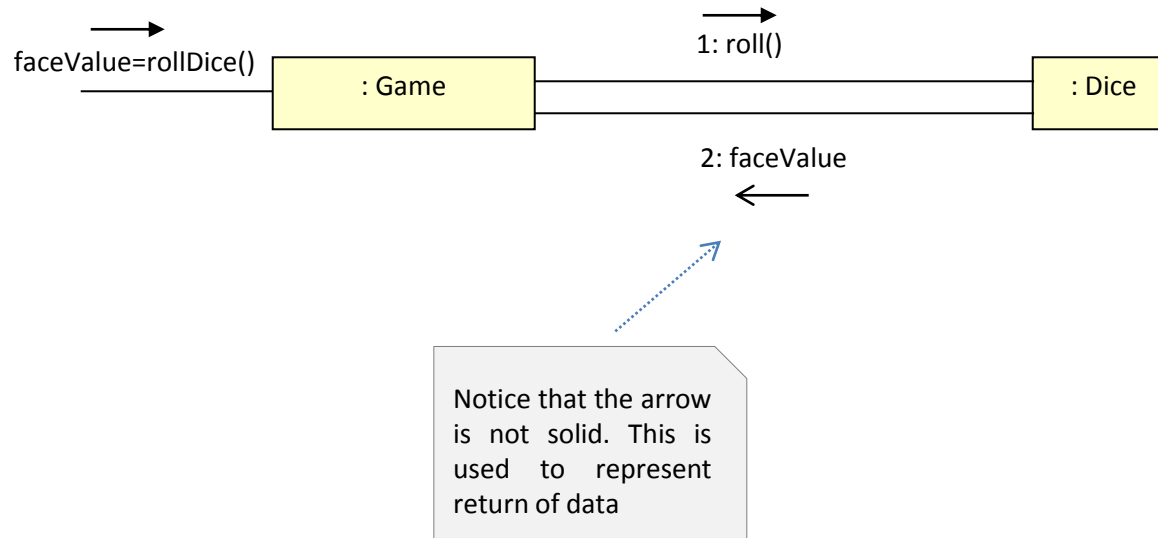


For the Snakes and Ladders case study, draw the communication diagram for roll the dice scenario.





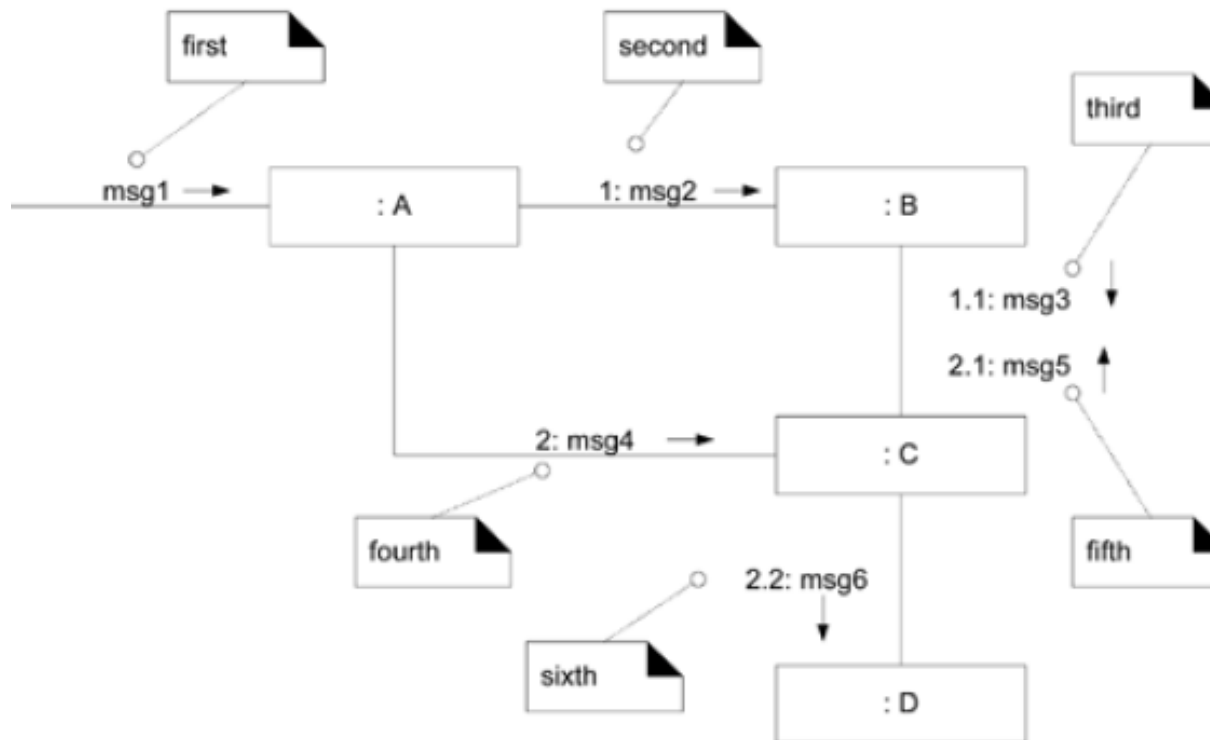
# Solution:



# Exercise: Communication Diagram



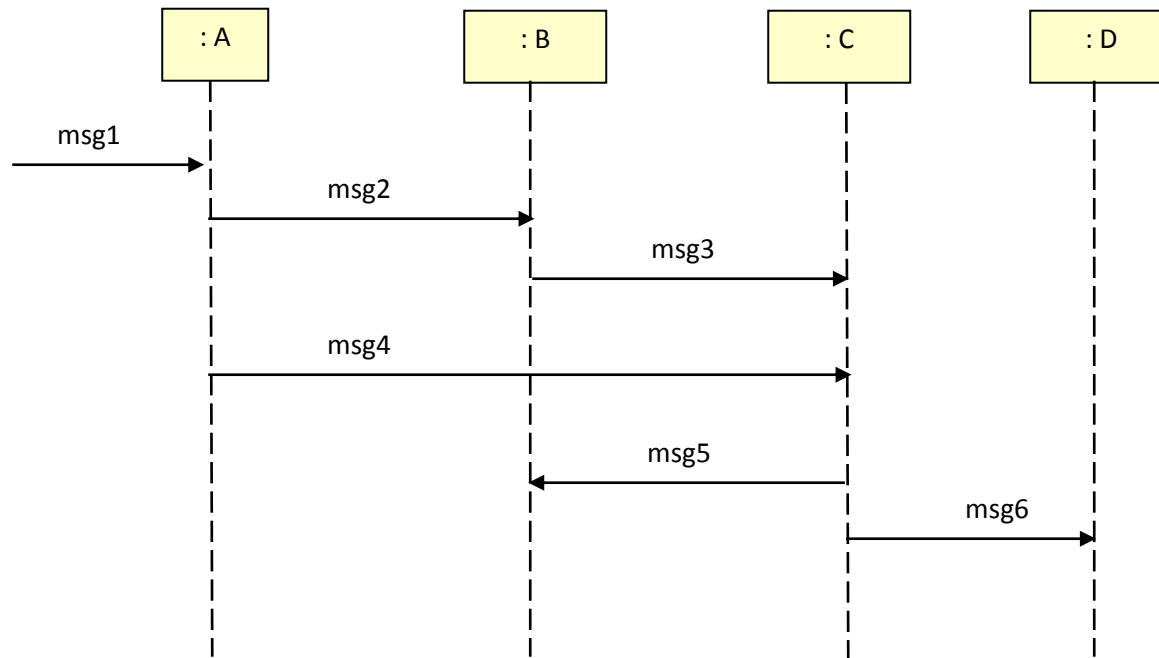
Convert the following communication diagram into sequence diagram:



Indexed numbering of messages in Communication Diagram

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

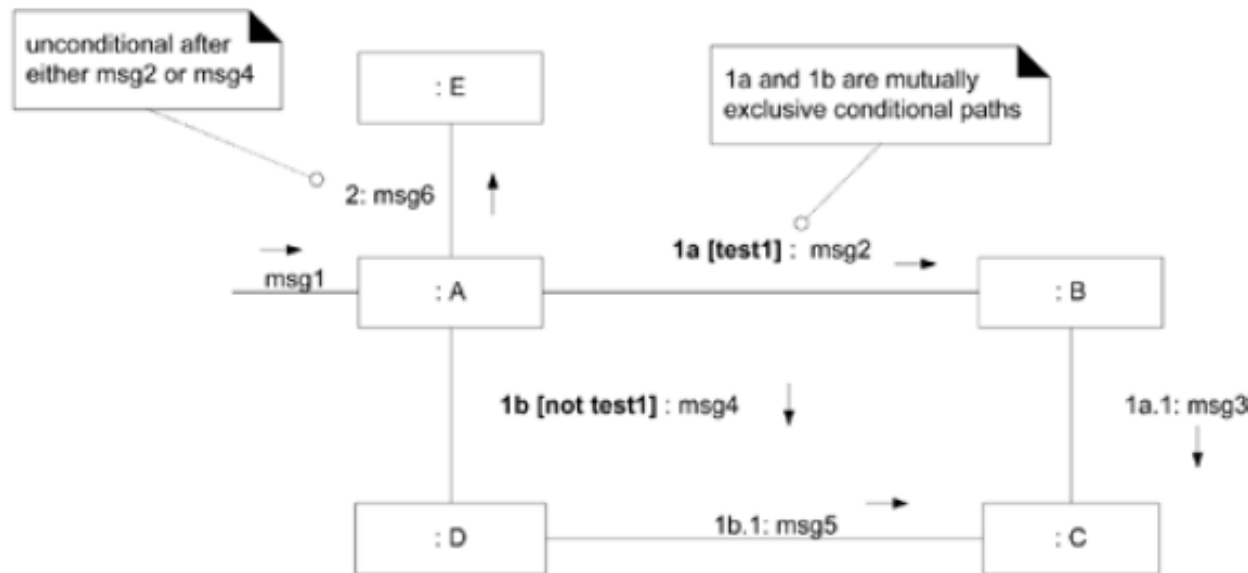
# Solution:



# Exercise: Communication Diagram



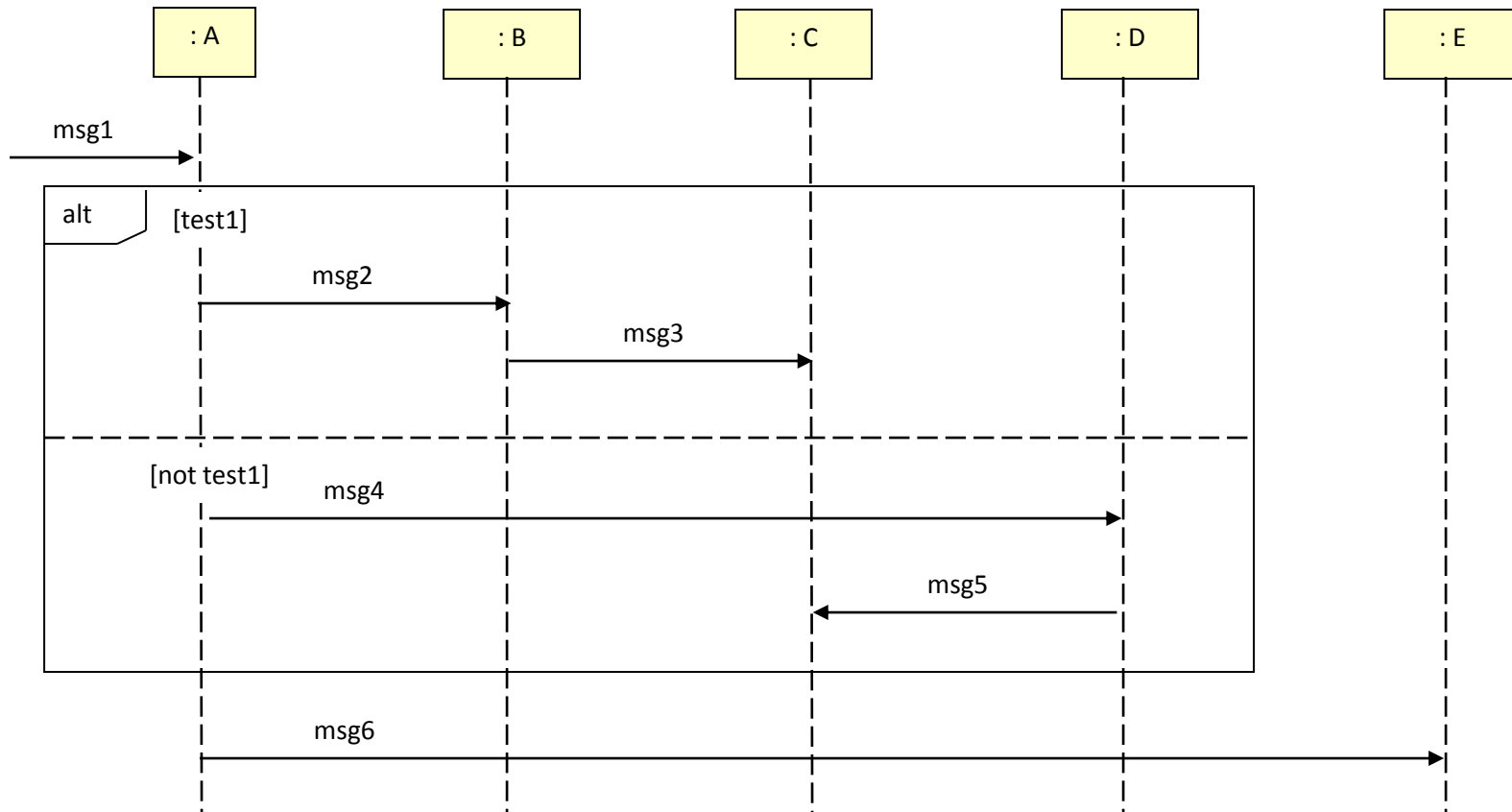
Convert the following communication diagram into sequence diagram:



Mutually exclusive conditional paths in Communication Diagram

Courtesy: Adapted from Applying UML and Patterns, Craig Larman, 3<sup>rd</sup> edition

# Solution:



# Plan ahead.....



Go through Lecture Videos:

- Module 4: State Transition Diagram
- Module 5: Visibility between Objects, Class Diagram

Agenda: Lecture 5

- State Transition Diagram
- Class Diagram