

UML Diagramming and Notation



Alif Finandhita

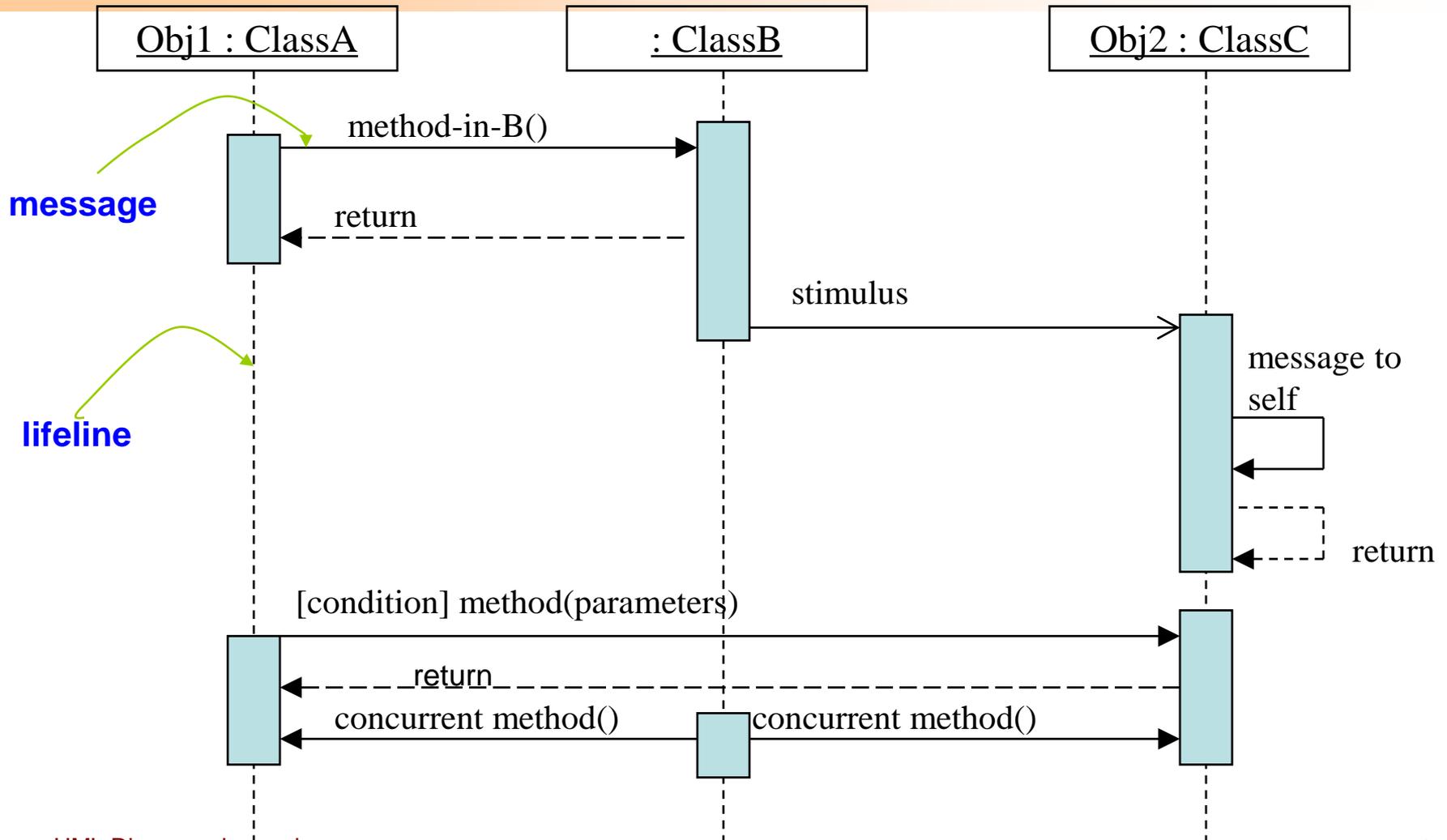
Teknik Informatika – Universitas Komputer Indonesia

Sequence Diagram Modelling

Sequence diagram

- A two-dimensional chart that describes collaborations
- Each sequence diagram describes a particular scenario
 - E.g., a sequence diagram can describe successful withdrawal operation
 - Another sequence diagram can describe failure of withdrawal operation due to insufficient funds
 - It is also possible to describe both scenarios in one sequence diagram

Sequence diagram: basic syntax



Sequence diagram – semantics (1)

- Rectangular boxes on the top indicate objects
- The dotted vertical line indicates the life line of the corresponding object
- Rectangular boxes on the dotted vertical line indicate the duration in which the corresponding object is active; the object is idle otherwise
- A solid arrow with solid arrowhead indicates a message
- A solid arrow with thin arrowhead indicates a stimulus
- A dashed arrow indicates return of control

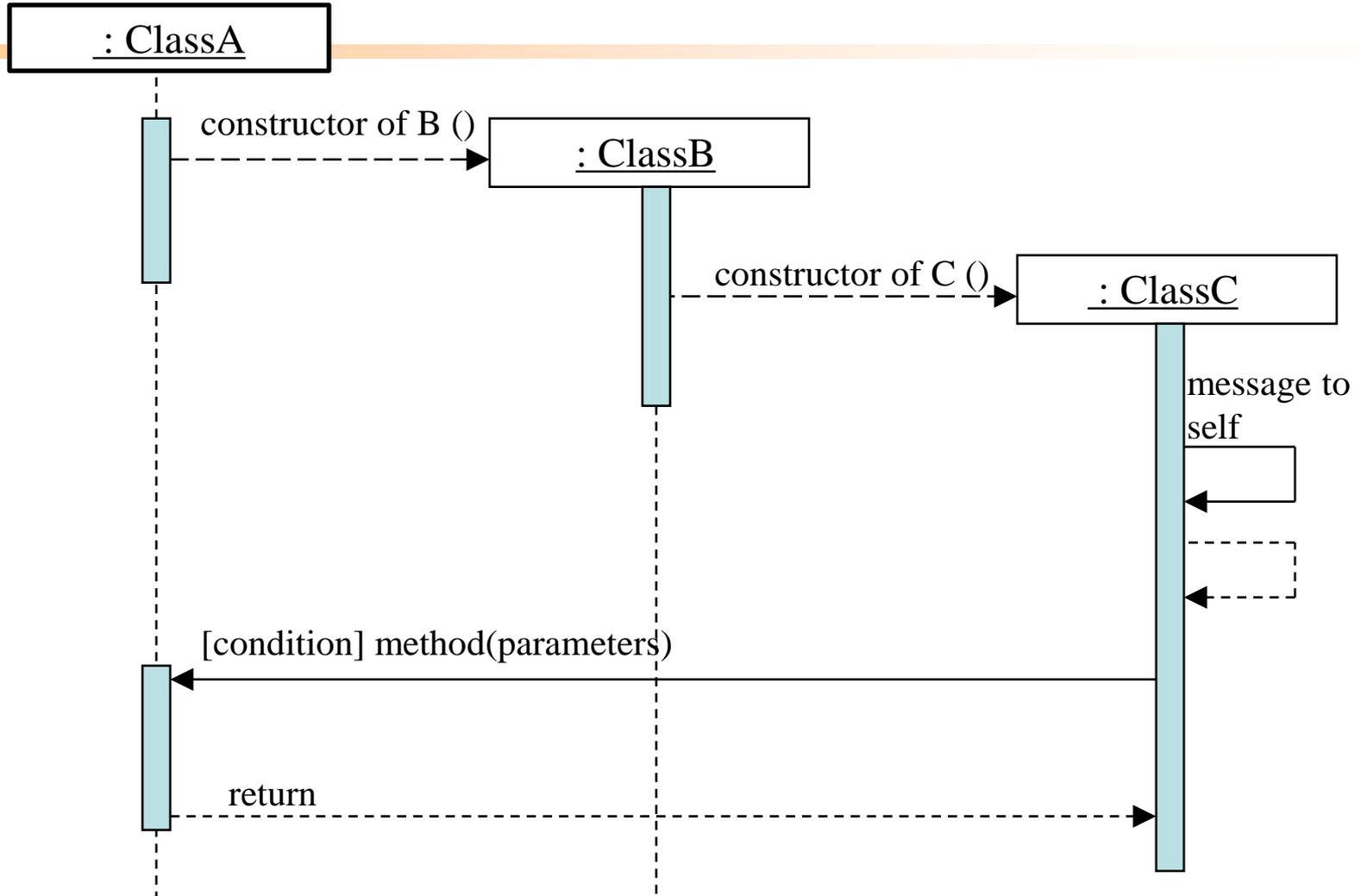
Sequence diagram – semantics (2)

- Messages/stimuli may be augmented with conditions
- Messages/stimuli can be concurrent
 - start from one object and send messages or signals to more than one object at the same time
- The vertical dimension indicates time axis
 - A message 'Y' placed below a message 'X' in a sequence diagram indicates that message 'X' is sent before message 'Y'

Sequence diagram – semantics (3)

- Objects with dashed vertical line for the entire diagram have lifeline for the entire scenario
 - These objects are assumed to be already created before this scenario starts and assumed to exist even after the scenario ends
- Objects with short life span within a scenario can be shown differently (see the next diagram)
- There is no ordering required among the placement of objects on the horizontal line
 - A designer may choose the ordering for the convenience of drawing the diagram

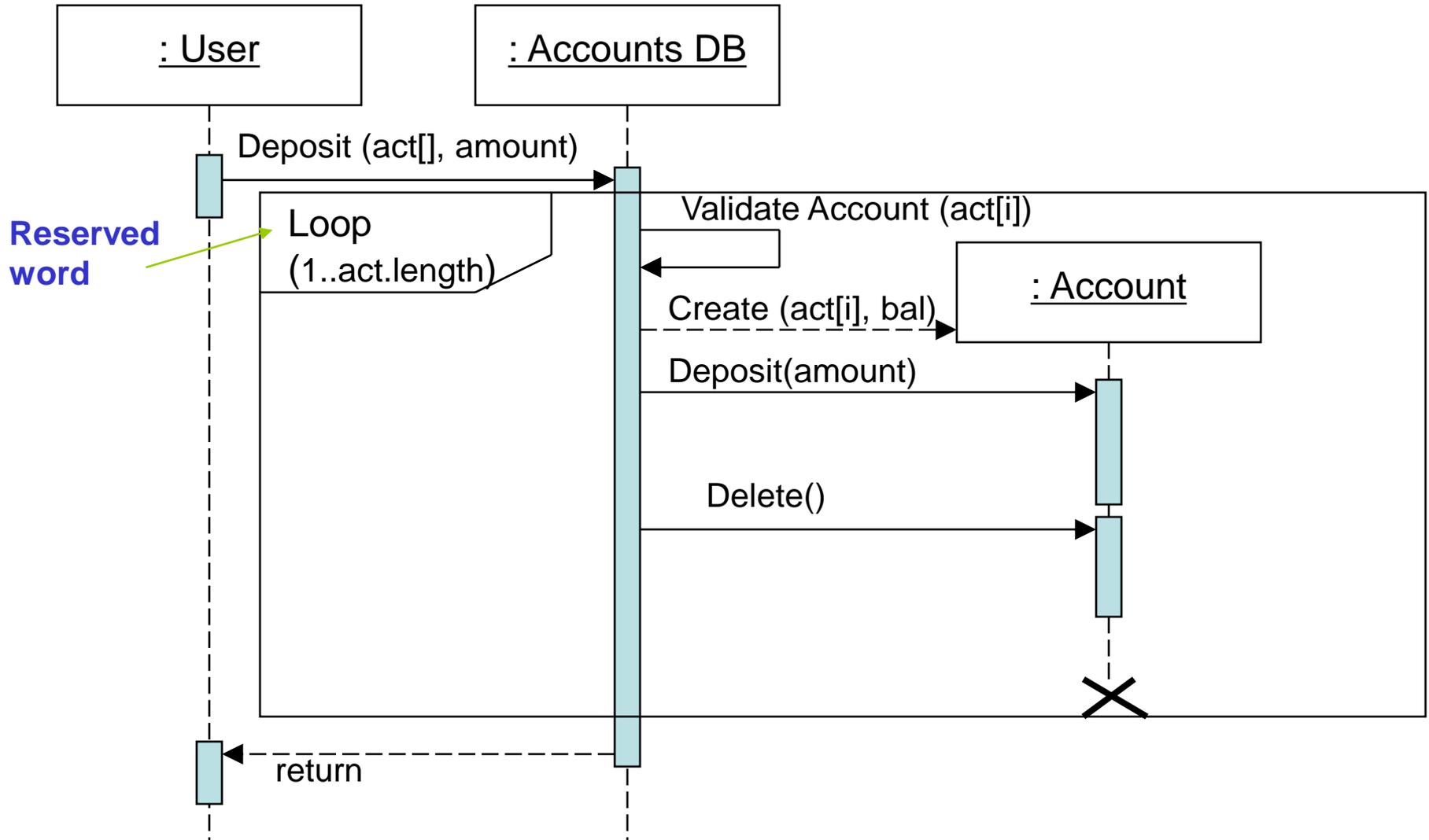
Sequence diagram – extended syntax



Sequence diagram – semantics (4)

- The creation of an object is shown by the vertical displacement of the object from the top of the diagram. The object is placed at the point of creation
 - See the creation of objects from class A and from class B
- The termination of an object can be shown by placing an “X” at the bottom of its life line
 - See the destruction of the object from class C

Sequence diagram – Specifying Loops



Depositing the same amount into several accounts

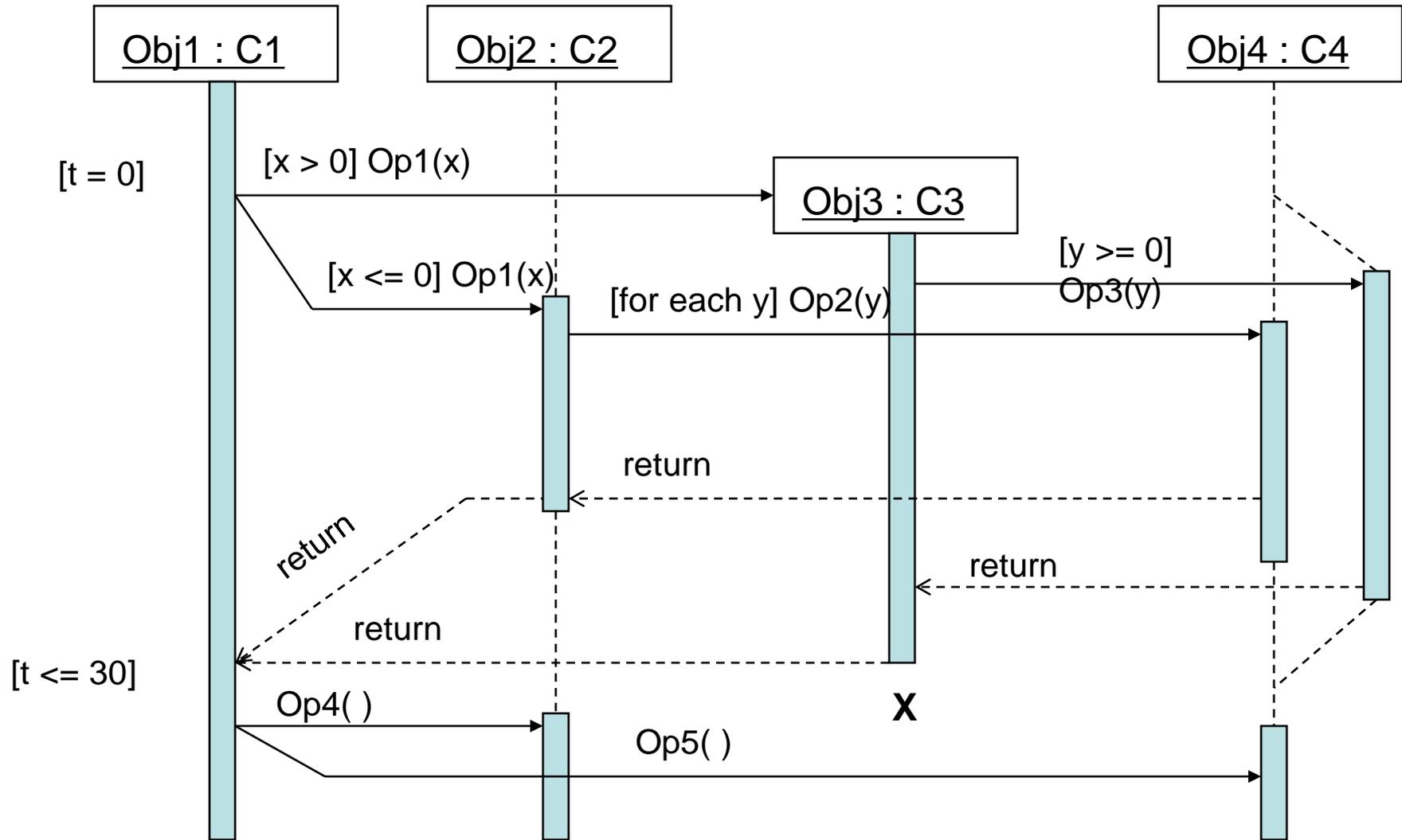
Concurrent messages / stimuli - semantics

- an object can send two messages / stimuli to two different objects at the same time
 - concurrent execution
 - there may be a condition on each of these messages / stimuli but they can be totally different
 - the tail end of these two messages / stimuli coincide at the originating object
 - the arrow heads may physically be on different horizontal levels but logically they are at the same time line (horizontal level)
 - see “Op4() and Op5()” in the diagram (next page)

Time-based messages / stimuli- semantics

- a message can be split into two, leading to two destination objects with mutually exclusive conditions
 - see “Op1(x)” in the diagram
- a lifeline can be split into a side track for a specific duration to indicate mutually exclusive situations
 - see “Obj4” in the diagram
- iterations can be specified using a condition at the beginning of a message / stimulus
 - see “Op2(y)” in the diagram
- timing constraints can be added onto the messages / stimuli or on the vertical time axis

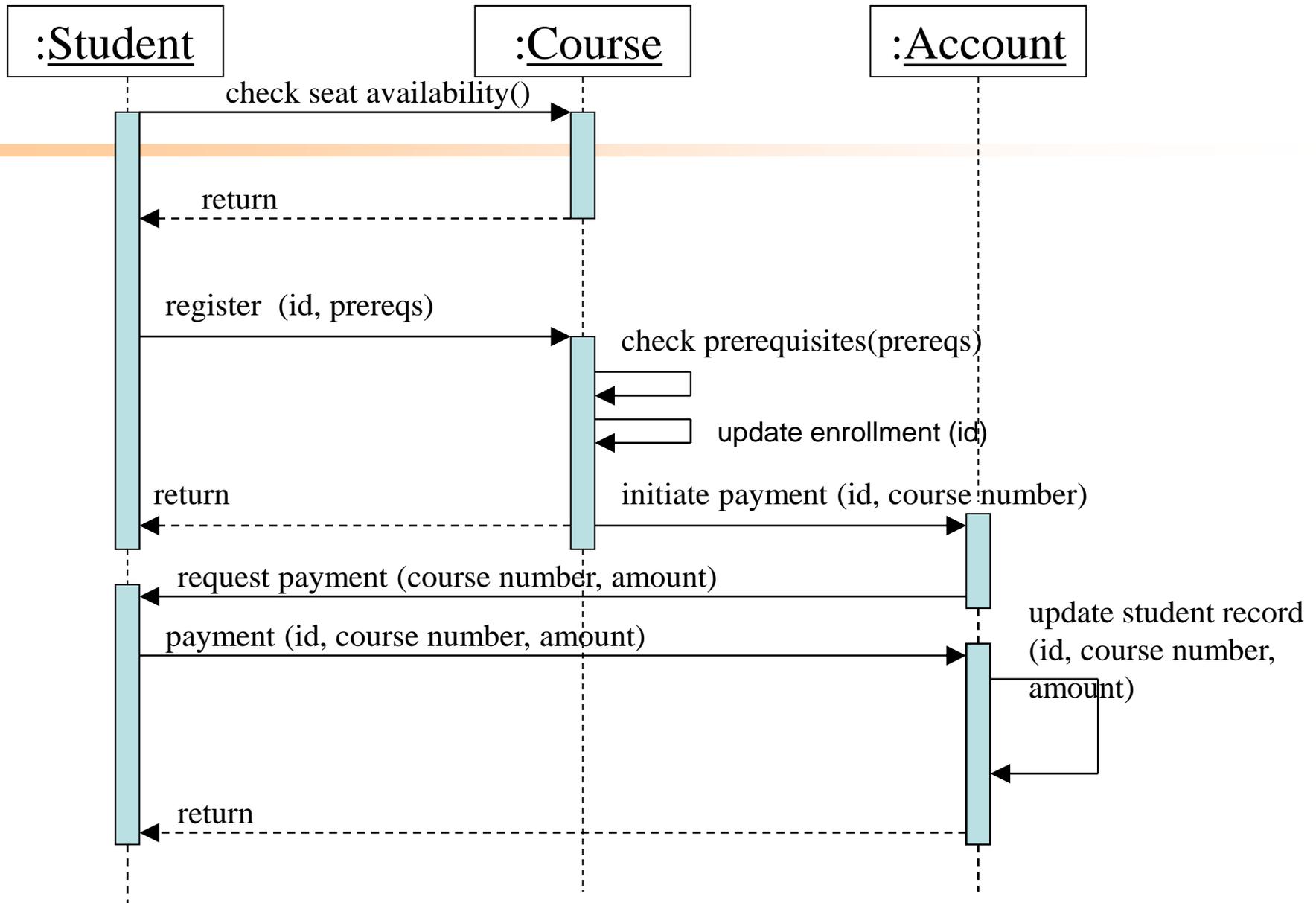
Concurrent and time-based messages and stimuli



Example – Course registration system

- Scenario

- A student object checks the availability of seats in a course. If available, it sends a message to register. The course object checks for the prerequisites first. Upon acceptance, the course object returns the message back to the student object and at the same time informs the account object to bill the student. The account object then communicates with the student to get the payment for the course.
- The following diagram shows a successful course registration process



References

1. Roger S. Presmann, Software Engineering, 6th edition.
2. Kendall, System Analysis and Design, 7th edition.
3. Ian Sommerville, Software Engineering, 8th Edition
4. PPT of Roger S. Pressman (chung and zheng)
5. PPT of Kendall
6. Saiful Akbar, Handouts PPL – ITB, 2011
7. Scott W. Embler, Elements of UML Style 2.0
8. Martin Fowler, UML Distilled 3, Third Edition