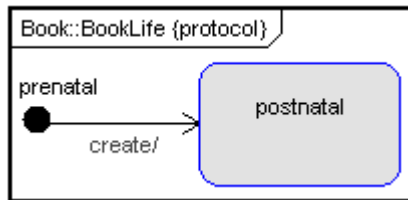
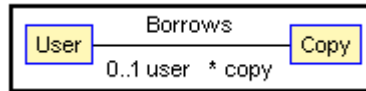
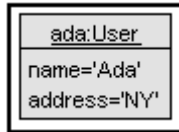
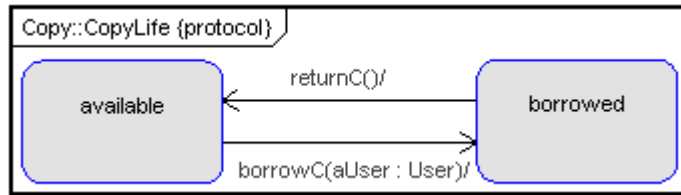


Ex A-01: For the stated notion mark one, two, three or four correct phrases.				
merkel // is a ...	State	Class	Object	Association
'Angela' // is a ...	ObjectId	String	Value	Datatype
merkel.init(...) // is a ...	Operation	Operation Call	Transition	Parameter
Profile::friends() // is a ...	Operation	Transition	Operation Call	Action
aFirstN // is a ...	Attribute	Parameter	Value	Use case
invitee // is a ...	Connection	Role	Status	Class
Associations are between ...	Objects	Classes	Transitions	Use Cases
Class diagrams define ...	Use Cases	Structure	Behavior	Transitions
State Charts detail ...	Structure	Behavior	Dynamics	States
Sequence diagrams show ...	Class Interaction	Object Interaction	Object Structure	Object Transitions
Transitions are between ...	Classes	Objects	State charts	States
Links connect ...	Concepts	Classes	Objects	Attributes

Ex A-02: Mark the correct diagram kind that uses elements of the shown items.

Interaction Communication Structure State Chart Sequence Class Behavior Object



Ex A-03: Mark the correct diagram kind that uses elements of the shown items.

Interaction

Communication

Structure

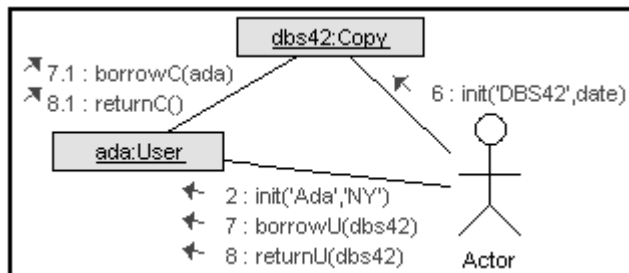
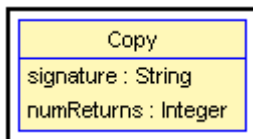
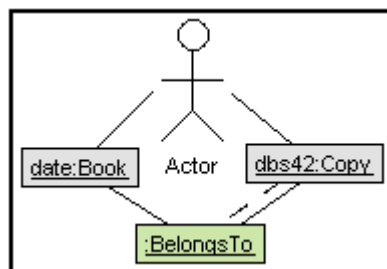
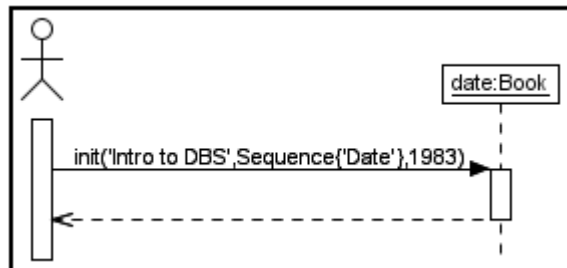
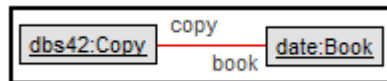
State Chart

Sequence

Class

Behavior

Object



Ex A-04: State the value of the OCL expression
in the object diagram in slide 14.

Value

may.lastN

may.invitee

putin.invitee.inviter

putin.inviter.invitee

commenting1.commented

commenting1.commenter

posting1.poster

posting1.commenter.inviter

Ex A-05: State whether the attempted calls are executable or not in the object diagram in slide 14.	executable	not executable	if executable, specify parameters (one choice)
merkel.init(...)			
merkel.accept(...)			
merkel.decline(...)			
putin.decline(...)			
putin.accept(...)			
putin.invite(...)			
may.decline(...)			
may.accept(...)			
may.invite(...)			
trump.decline(...)			
trump.accept(...)			
invite.invite(...)			

Ex A-06: Tick OCL usage in the diagrams from the current teaching unit.	Query	Class invariant	State invariant	Operation precond.	Operation postcond.	Transition precond.	Transition postcond.	Operation definition
Class dia.								
Object dia.								
Sequence dia.								
Communication dia.								
State chart dia.								

Ex A-07: Construct 4 permutations of the command sequence in slide 35 each leading to same object diagram in slide 14.	Permutation 1	Permutation 2	Permutation 3	Permutation 4
!create merkel,putin,trump:Profile				
!merkel.init('Angela','Merkel','muddi')				
!putin.init('Vladimir','Putin','crab')				
!trump.init('Donald','Trump','theDonald')				
!putin.invite(merkel)				
!trump.invite(putin)				
!putin.decline(trump)				
!merkel.accept(putin)				
!p:=merkel.publish('BMW, we have a problem')				
!create may:Profile				
!may.init('Theresa','May','motherTheresa')				
!putin.comment(p,'May the Donald be with you')				
!may.invite(merkel)				

Ex A-08: Bring the OCL expressions and the explanations into order (referring to the object diagram on slide 14).	
Who are the friends of profile with initials 'AM'?	Set{Friendship3.inviter, Friendship3.invitee}
How is the union of the inviters and invitees of the profile with user name crab calculated?	Profile.allInstances->select(p p.inviter->notEmpty)
Who is the poster of the posting underlying the only commenting object?	trump.friends().friends()
Which profiles received invitations?	Commenting1.commented.poster
Which profiles issued invitations?	Friendship.allInstances->select(fs fs.status='accepted' or fs.status=declined)
Which friendships are accepted or declined?	merkel.friends()
Who are inviter and invitee of the pending friendship?	Profile.allInstances->select(p p.invitee->notEmpty)
Who are the friends of the friends of profile with initials 'DT'?	putin.invitee->union(putin.inviter)

Ex A-09: List the 12 most important notions from the current teaching unit and define them in an informal, verbal manner.	
---	--