Q1: The advantages of the composite pattern:

To treat individual components and their composite uniformly.

New type of components can be added easily.

What is composite pattern? <https://www.youtube.com/watch?v=mp5lwolO-wM>

Basically, in OOP is it known as “has-a” relationship.

The goal of composite pattern is to get the objects to represent a tree pattern.

If an object contains a list of objects, that object is know as a composite. Also, this composite object can contain other composite objects. If you can perform an operation on a leaf node, than we need to be able to perform that same operation on this composite node.

Example **wrong** answers:

To provide clean separation between control logic and display logic.

To provide a new interface to existing legacy components.

Monitoring objects can be added or removed with freedom.

Minimal coupling between subjects and monitoring objects.

Q2: Which of the following expressions is correctly typed in Event-B?

**This is correct:**



Not sure about the 1-1 part. The empty set is fine because it is representing the codomain.

**These are not correct:**



Empty set cannot be a domain.



Empty set cannot be a domain.



You cannot have a set within a set if its not within a function.

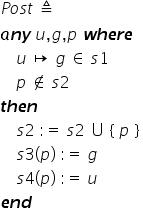
Q3:   
We consider a simple social network (*SocNet*) allows people to join the system as users. A user can create a new group which they then become the owner of. A user can request to join a group which has to be approved by the group’s owner. Users can post to a group if they are a member of that group.

The Event-B model of *SocNet*contains variables as follows:

* *user*representing the registered users,
* *group*representing the groups that have been created,
* *post*representing the posts that have been created,
* *joined*representing the groups a user has joined,
* *owner*representing the owner of a group,
* *sender*representing the sender of a post,
* *dest*representing the group that a post is sent to.

Part 1:

The Event-B model of *SocNet*contains an event modelling a user posting to a group. In the following version of the event, *s1* to *s4* are placeholders for machine variables:



Possible answers:

|  |
| --- |
| A.  *dest* |
| B.  *post* |
| C.  *sender* |
| D.  *joined* |

**Correct answers:**

A

joined

B

post

C

dest

D

Sender

Part 2:

Match the following variable names with the appropriate constraint:









Possible answers:









Correct answers:









Part 3:

The Event-B model of *SocNet*includes the following invariant:



Which **one**of the following best describes this invariant:

The CORRECT answer:

The owner of a group must be a member of that group

The INCORRECT answers:

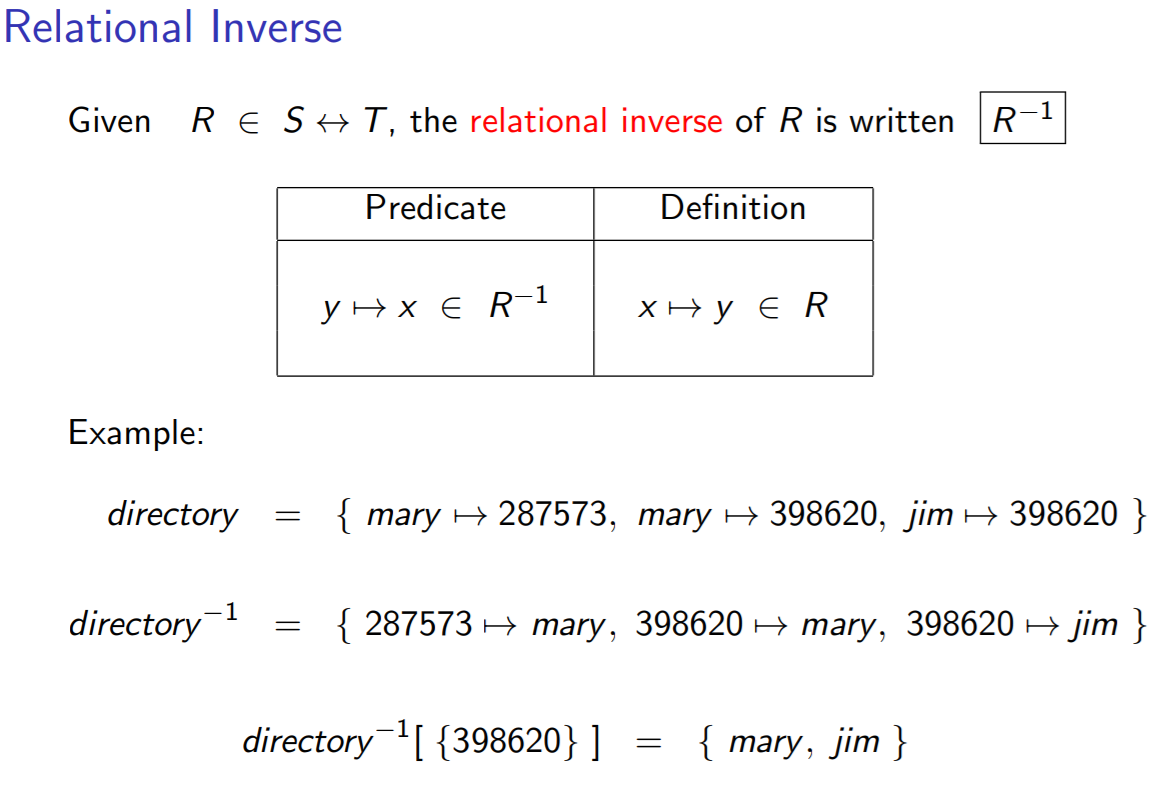
The users who have joined must be non-owners of the group.

Duh

There are fewer owners than joiners in a group.

Nope

## The opposite owner must have joined the group Nope



**LECTURE RELATIONS AND FUNCTIONS (CONTINUED)**

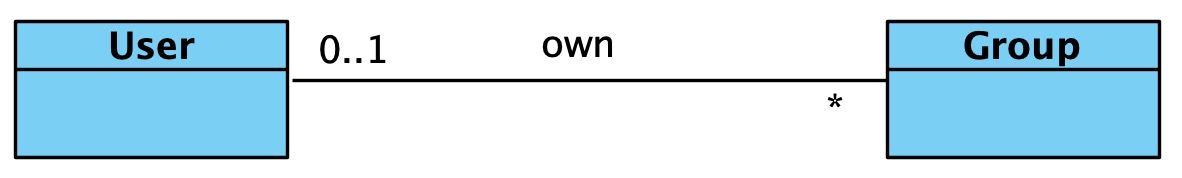
Q4:

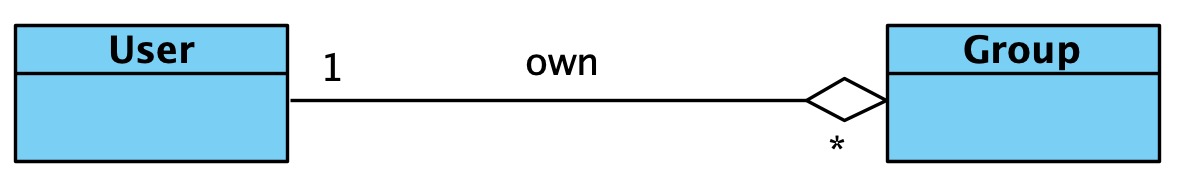
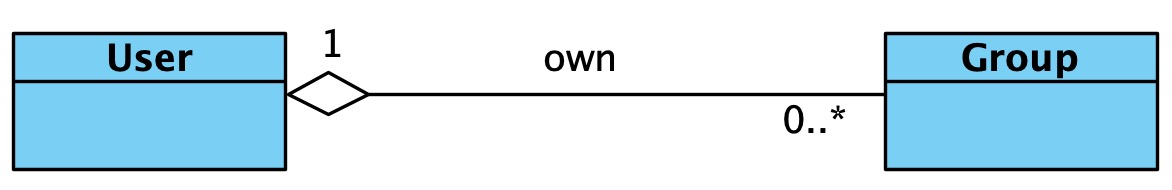
Part 1:

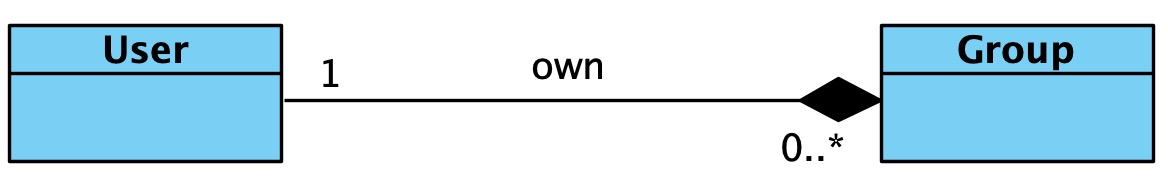
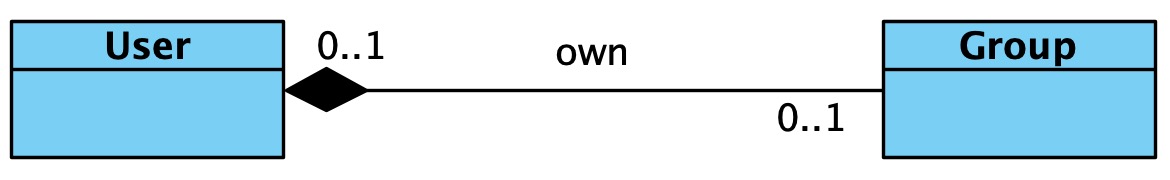
Consider a social network (SocNet) that allows people to join the system as users. A user can set up a new group which they then become the owner. A user can request to be a member of a group. Two users can become friends with each other. A post by a user (the creator) is accessible only to the creator’s friends. A post can be a textual post or an image post. Furthermore, image posts can have different effects applied to them.

The system will use the following classes: User, Group, Post, TextPost and ImagePost.

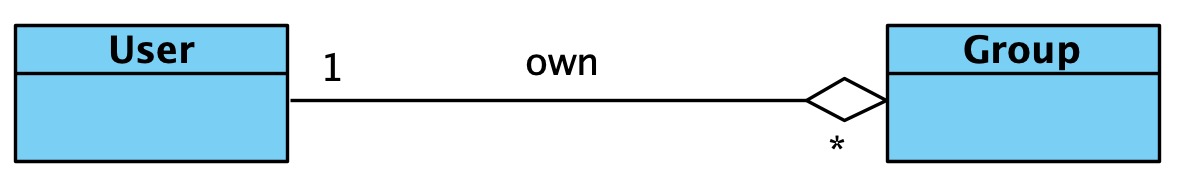
Which of the following is the most appropriate to model the own relationship between User and Group?



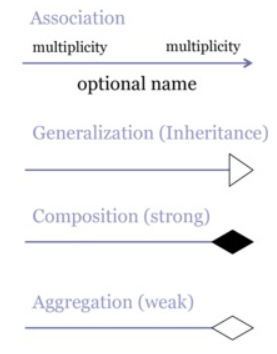




The correct answer is:



**What do these arrows mean?**



Explained more in depth in next question.

Part 2: <https://www.guru99.com/association-aggregation-composition-difference.html#1>

Check out table at the end of the website.

The system will use the following classes: User, Group, Post, TextPost and ImagePost.

Which of the following is the most appropriate to model the own relationship between User and Group?

CORRECT answer: **aggregation**

Subtype of an association.

TLDR: an object of one class can own or access objects of another class.

INCORRECT answers:

Inheritance

easy

Multiplicity

I mean that is just a wrong answer.

|  |  |  |
| --- | --- | --- |
| **Multiplicity** | **Option** | **Cardinality** |
| **0..0** | **0** | Collection must be empty |
| **0..1** |  | No instances or one instance |
| **1..1** | **1** | Exactly one instance |
| **0..\*** | **\*** | Zero or more instances |
| **1..\*** |  | At least one instance |
| **5..5** | **5** | Exactly 5 instances |
| **m..n** |  | At least m but no more than n instances |

Association

Objects can be connected or associated with other objects of the system.

Composition

Subtype of aggregation. But it is stricter. That means that if two objects are connected by aggregation, if you delete one of them, the other cannot function aswell.

Q5:

Axiom Assumptions about sets and constants of a model

Invariant Properties of a model variables that should be preserved by events

Guard Condition under which an event may be executed

Event Specifies changes to the variables of a model

|  |
| --- |
|  |
|  |