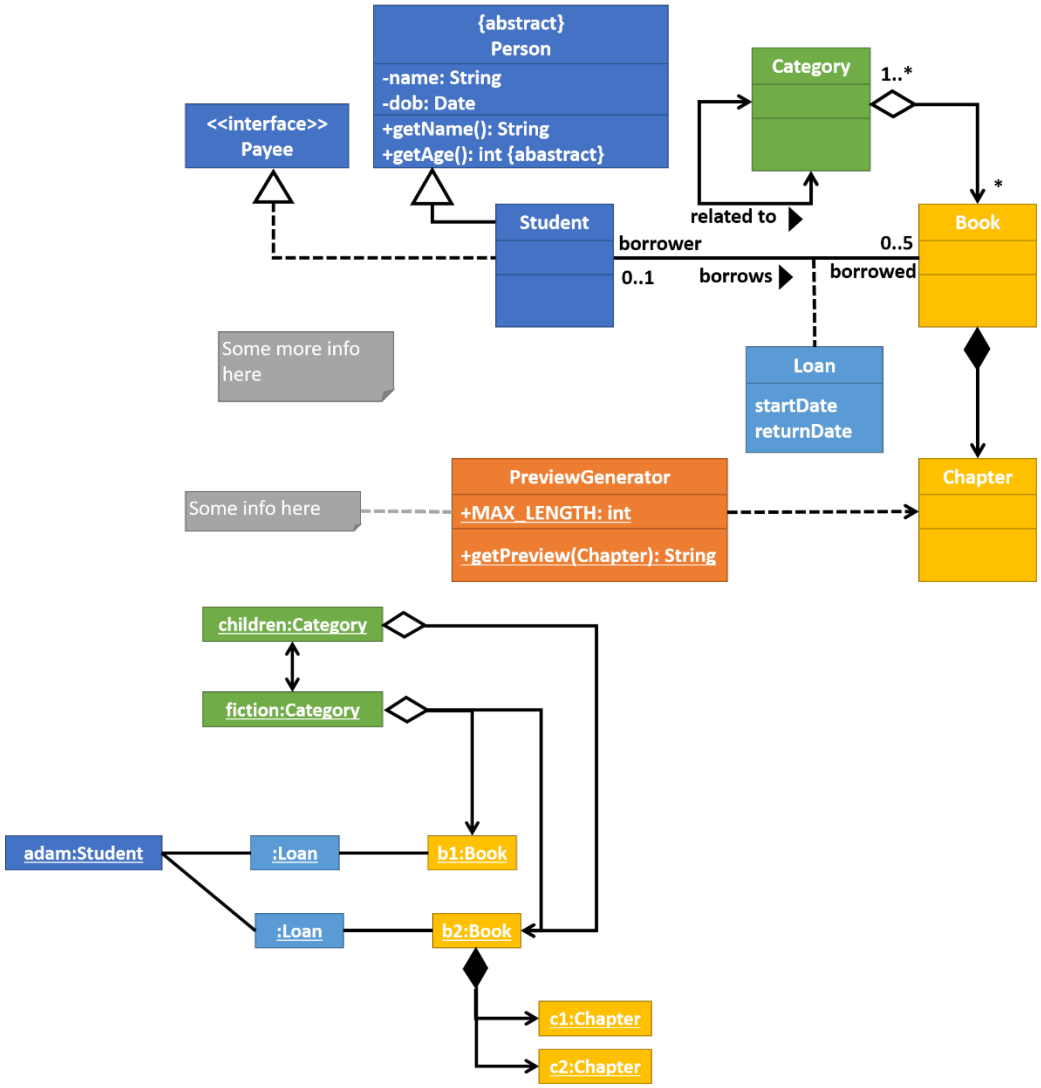
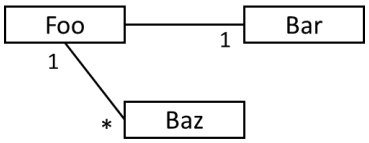
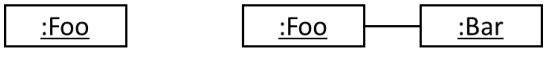
1. Discuss the class diagram and validate if the object diagram corresponds to the class diagram given

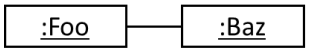


2. Which object diagram corresponds to the given class diagram?





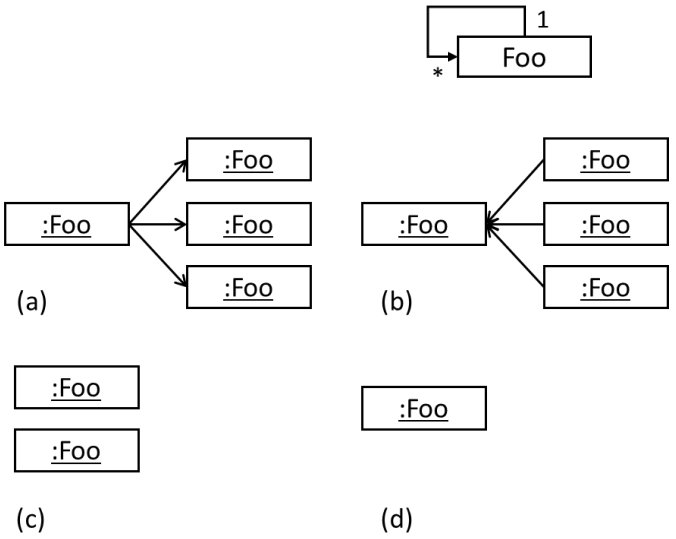
(a) (b)



(c)

(b) Others don't satisfy the mandatory association between Foo and Bar

3. Which object diagram corresponds to the given class diagram?

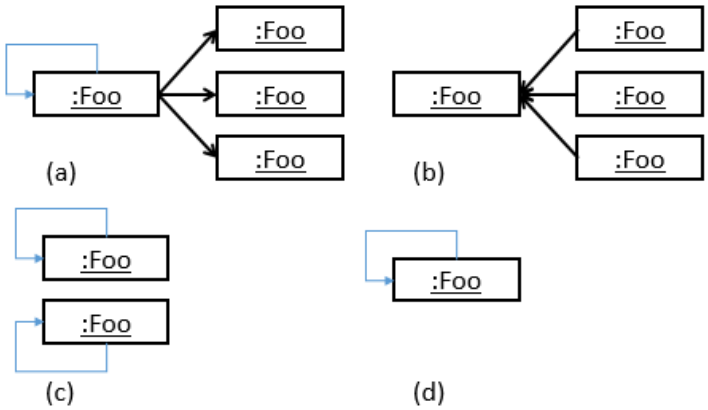


(Clarification) The question in the given form doesn't have any correct answer.

The way to interpret this class diagram is: One Foo object can associate with (for the sake of understanding only: "point to") any number of Foo objects. Every Foo object must be associated with one Foo object (for sake of understanding only: "pointed to by"). Hence, none of the options satisfy the "pointed to by" condition.

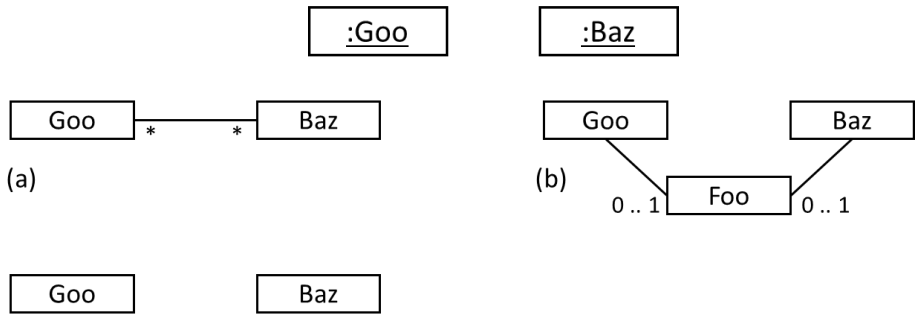
We can have free standing Foo objects only if they associate with themselves.

When the object diagrams are updated as shown below (notice the associations marked in blue), the answer is (a), (c) and (d)



In many cases, for self-associations, roles are specified. Using the information about the roles make it easy to identify the object structure. However, in this particular case, there is no information about the role, which makes it a bit tricky.

4. Which class diagram corresponds to the given object diagram?



(a), (b) and (c).

(a) satisfies because the association has multiplicity *, so we can have 0 associations between Goo and Bar objects

(b) satisfies trivially, as there are 2 independent classes and there are 2 independent objects.

(c) satisfies due to the fact that the association between Foo and Goo/Baz are optional (0..1 => optional); in the object diagram they are not present.