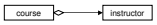
Discussion Slides

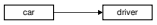
Deeper

- Give a few examples of a unidirectional reference?
  - *If an appointment object contains, within it, a date object.*

  - *If a course object includes a pointer to the object for the associated instructor.*



  - *A car object is (in some way) associated with the object for the current driver.*



Deeper

- What is the difference between aggregation and composition?
  - *Composition is aggregation with strong ownership.*
  - *In simple aggregation, the whole includes a reference to the part, but the part may outlive the whole and be referenced by other objects.*
  - *In a composition relationship, the part is owned by and ends with the whole.*

Deeper

- Give a few examples of bidirectional associations?
  - *One course, one roster, and one grade book might all be mutually associated with one-another.*
  - *The association could be implemented with pointers, a shared index, or a higher level object that contained or referenced the others. In this last case, the higher level object might (if it has properties and/or methods) be an instance of an association class.*

Deeper

- What makes UML class diagrams good for sketching out a design?
  - *The language is fairly complete, but you are free to leave out as much as you want.*
  - *You can list only the methods and properties that are relevant to the current picture.*
  - *You can show only the relationships that are relevant to the current picture.*

Deeper

- How do dependencies differ from associations?
  - *Associations are structural relationships between objects (e.g. one contains or points to the other).*
  - *Dependencies are relationships between classes, where one class uses the information or services of another (e.g. being derived from or making calls to).*

Deeper

- Why is the same notation used for constraints and comments?
  - *Both provide supplementary information about the object, attribute or method.*
- How might a constraint differ from a comment?
  - *A constraint might be part of a declaration (e.g* **readonly***) or an assertion that could turn into code.*

Deeper

- What appears in class diagrams, but wouldn't appear in an object diagram?
  - *Class inheritance/derivation information has no place in an object diagram.*
  - *Type declarations might be present (as reminders) on attributes, but method declarations make very little sense.*

Deeper

- Would it make sense to show interface providers on an object diagram?
  - *Definitely. If the interface provider was bound to an object at run time, it could be very important to note which implementation was being used.*

Deeper

- This looks very much like a class or object model. Why isn't it one?
  - *It looks graphically very similar, but the meanings of the fields in each screen box do not match the (well defined) meanings of the corresponding fields in a class diagram.*
  - *UML is not a drawing style. It is a language, with very specific syntax and vocabulary. If you stray from that definition, you are no longer UML.*

Deeper

- What connects UML behavioral models (e.g. an activity diagram) with UML structural models?
  - *An activity diagram can be put inside of a class diagram to show the algorithm associated with the implementation of an interface.*

Deeper

- How do UML use case diagrams integrate with other structural or behavioral UML models?
  - *Activity diagrams show algorithm implementations, but how do we associate these with use case action ovals?*
  - *Use case diagrams might indicate domain objects in the user's world view, but how do we map these into the solution domain objects on class diagrams?*