Whatisastereotype? What does it mean? For what can it be useful?

Could you please map this signature to a class diagram?

How are system states and object diagrams related? What is visibility good for? Where is it used?

What's the difference between "aggregation" and "composition"?

Please un-abbreviate all abbreviations in this OCL expression.

For what purposes are class diagrams useful?

What's "multiplicity"? How did we treat them semantically?

Could you explain this class diagram with associations?

What's the purpose of the X diagram?

Please formalise this constraint in OCL.

Do you have any ideas about a system state satisfying this constraint?

Please explain this OCL constraint. Is it well-typed?

Which annotations of an association arrow are (semantically) relevant?

How do Basic Object System Signatures relate to UML class diagrams?

Capabilities for the following tasks/questions:

Educational Objectives:

In what modelling language is UML modelled?

What is an object diagram? What are object diagrams good for?

When is a set of OCL constraints said to be consistent?

What does it mean that an OCL expression is satisfiable?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modellling language is it modelled)?

What is a UML model (in what modelling language is it modelled)?

What is the purpose of the X diagram?

What is the principle behind that?

What is a UML model (our definition)? What does it mean?

What kind of diagrams does UML offer?

What is the purpose of the X diagram?

What are the intentions of UML?

What is UML (definitely)? Why?

What is UML (definitely)? Why?

How does UML relate to programming languages?

What is UML (definitely)? Why?

What are the benefits people see in meta-modelling?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (in what modelling language is it modelled)?

What is meta-modelling?

What benefits do you see in meta-modelling?

What is class diagram in the context of meta-modelling?

What is a UML model (in what modelling language is it modelled)?

What is an object diagram? What are object diagrams good for?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?

What is a UML model (in what modelling language is it modelled)?

What is a UML model (our definition)? What does it mean?
Wrapup: Modelling Structure

Lecture 9:

- Educational Objectives:
  - Capabilities for following tasks/questions.
  - What are the purposes of modelling guidelines? (Example?)
  - When is a class diagram a good class diagram?
  - Discuss the style of this class diagram.

Lecture 20 & 21:

- Educational Objectives:
  - Capabilities for following tasks/questions.
  - What's the effect of inheritance on System States?
  - What does the Liskov Substitution Principle mean regarding structure?
  - What is the subset, what the uplink semantics of inheritance?
  - What's the idea of Meta-Modelling?

Wrapup: Modelling Behaviour, Constructive

Lecture 10:

- Educational Objectives:
  - Capabilities for following tasks/questions.
  - What's the difference between reflective and constructive descriptions of behaviour?
  - What's the Basic Causality Model?
  - What does the standard say about the dispatching method?
  - What is (intuitively) a run-to-completion step?

Lecture 11:

- Educational Objectives:
  - Capabilities for following tasks/questions.
  - Please model the following behaviour. (And convince readers that your model is correct.)
  - What is: trigger, guard, action?
  - Please unabbreviate this abbreviated transition annotation.
  - What is an ether? Example? Why did we introduce it?
  - What's the difference: signal, signalevent, event, trigger, reception, consumption?
  - What's a system configuration?
  - When is an object stable? (Intuitively, formally?)

Lecture 12 & 13:

- Educational Objectives:
  - Capabilities for following tasks/questions.
  - What is a transformer? Example? Why did we introduce it?
  - What is a re-uses semantics? What of the framework would we change to go to a non-re-uses semantics?
  - What labelled transition system is induced by a UML model?
  - What is: discard, dispatch, commence?
  - What's the meaning of stereotype "signal, env"?
  - Does environment interaction necessarily occur?
  - What happens on "division by 0"?

Lecture 14:

- Educational Objectives:
  - Capabilities for following tasks/questions.
  - What is a step (definition)? Run-to-completion step (definition)? Microstep (intuition)?
  - Do objects always finally become stable?
  - In what sense is our RTC semantics not compositional?
Wrapup: Modelling Behaviour, Constructive

Lecture 21:

Educational Objectives:
Capabilities for following tasks/questions.

- What's a history state? Deep vs. shallow?
- What is junction, choice, terminate?
- What is the idea of deferred events?
- What is a passive object? Why are passive reactive objects special? What did we do in that case?
- What's a behavioural feature? How can it be implemented?

Wrapup: Modelling Behaviour, Reflective

Lecture 1: Motivation and Overview
Lecture 2: Semantical Model
Lecture 3: Object Constraint Language (OCL)
Lecture 4: Object Diagrams
Lecture 5: Class Diagrams I
Lecture 6: Type Systems and Visibility
Lecture 7: Class Diagrams II
Lecture 8: Class Diagrams III
Lecture 9: Class Diagrams IV
Lecture 10: Core State Machines I
Lecture 11: Core State Machines II
Lecture 12: Core State Machines III
Lecture 13: Core State Machines IV
Lecture 14: Core State Machines V, Rhapsody
Lecture 15: Hierarchical State Machines I
Lecture 16: Hierarchical State Machines II
Lecture 17: Live Sequence Charts I
Lecture 18: Live Sequence Charts II
Lecture 19: Live Sequence Charts III
Lecture 20: Inheritance I
Lecture 21: Deferred Events, Behavioural Features, Inheritance II
Lecture 22: Meta-Modelling, Inheritance III
Lecture 23: Wrapup & Questions

Wrapup: Modelling Behaviour, Reflective

Lecture 17, 18, & 19:

Educational Objectives:
Capabilities for following tasks/questions.

- Is each LSC description of behaviour necessarily reflective?
- There exists another distinction between "inter-object" and "intra-object" behaviour. Discuss in the context of UML.
- What does this LSC mean?
- Are these UML model's state machines consistent with the interactions?
- Please provide a UML model which is consistent with this LSC.
- What is: activation (mode, condition), hot/cold condition, pre-chart, cut, hot/cold location, local invariant, legal exit, hot/cold chart, etc.

Wrapup: Inheritance

Lecture 1: Introduction
Lecture 2: Semantical Model
Lecture 3: Object Constraint Language (OCL)
Lecture 4: Object Diagrams, Class Diagrams I
Lecture 5: Class Diagrams I
Lecture 6: Type Systems and Visibility
Lecture 7: Class Diagrams II
Lecture 8: Class Diagrams III
Lecture 9: Class Diagrams IV
Lecture 10: Core State Machines I
Lecture 11: Core State Machines II
Lecture 12: Core State Machines III
Lecture 13: Hierarchical State Machines I
Lecture 14: Hierarchical State Machines II
Lecture 15: Hierarchical State Machines III
Lecture 16: Methods, Live Sequence Charts II
Lecture 17: Live Sequence Charts II
Lecture 18: Live Sequence Charts III, Inheritance I
Lecture 19: Inheritance II, Meta-Modelling I
Lecture 20: Meta-Modelling II, Inheritance III
Lecture 21: Wrapup & Questions

Wrapup: Inheritance

Lecture 20 & 21:

Educational Objectives:
Capabilities for following tasks/questions.

- What's the effect of inheritance on LSCs, State Machines, System States?
- What's the Liskov Substitution Principle?
- What is commonly understood under (behavioural) sub-typing?
- What is the subset, what the uplink semantics of inheritance?
- What is late/early binding?
- What's the idea of Meta-Modelling?
Hmm...

• Open book or closed book?