Student Online Teaching Advice Notice

The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.

Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.

Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.

Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.

Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's policies and procedures.

Further information on data protection and best practice when using videoconferencing software is available at https://www.tcd.ie/info_compliance/data-protection/.

CSU22041: Information Management I

Information Representation and Querying Using eXtensible Markup Language(XML) and XQuery

… an **art of making information accessible**.

2020-2021

Gaye Stephens gaye.stephens@tcd.ie

From Introductory Lecture

Core Concepts

ORGANISATION- How data is represented/associated

METADATA Data about what the data is

ACCESS How to access the data efficiently

Moving on

- Videos and supplementary material will focus on
  - XML- Extensible Markup Language
  - DTD – Document Type Definition
  - Xpath – to navigate a path through the XML document
  - Xquery – to Query the XML documents

- Second part of the assignment
  - Continue in the groups that you were in for the first part of the semester.
  - The group work for this second part may be less collaborative on an ongoing basis.

- Plan for the remaining 5 weeks of the teaching semester

- Introduce transitioning from UML to XML

Suggested Plan for the remaining 5 weeks of teaching in Semester 1

Week	Suggested Student Activities	Videos	Other Supporting Material	Interaction with Lecturer and TAs*
Beginning 16th Nov	1. Review videos 2. Read the assignment sheet 3. Take a look at the XML & DTD exercises 4. As a group begin to create XML documents and DTDs based on the classes chosen from your UML class diagram.	1. Going from UML to XML 2. Parts of an XML Document 3. Document Type Definition(DTD)	1. UML-XML Example(Video) 2. XML & DTD exercises	The plan for this week is for students to use their time to individually review the uploaded material and as a group to begin planning for the second part of the assignment. To answer any queries or provide clarification Gaye Stephens will be available online in the blackboard course room during the timetabled F2F sessions for PODs. **If you don't have any questions at this point there is no need to join the session.**
Beginning 23rd Nov	1. Review Videos 2. Continue to develop your XML and DTD documents. 3. Begin to create XQueries to support some of your use cases. 4. Try out some XQuery exercises using XBase and xml files	1. XPath 2. XQuery	1. XPath & XQuery exercises and supporting files can be found under XML Exercises tab on blackboard 2. XML & DTD exercise Solutions 3. Assignment Review (video)	Discuss progress on your XML and DTD files with TAs in your blackboard groups during the hour allocated to your assignment group.
Beginning 30th Nov	1. Continue to refine your XQueries, XML documents and DTDs 2. Check out solutions to the XPath Xquery exercises.		1. XPath & XQuery exercise Solutions (Video)	Show progress on your XML, DTD and Xqueries to TAs in your blackboard groups during the hour allocated to your assignment group.
Beginning 7th Dec	1. Show your queries running to your TA **2. Submit XML files and XQueries to Blackboard by 11th December.**	Exam Preparation		Present queries to TAs in your blackboard groups during the hour allocated to your assignment group.
Beginning 14th Dec	Finalise your report for submission **Submit XML Report by 18th December**	1. Linked Data and RDF 2. Wrap Up examples		Q+A session for all on Thursday 17th December 11-12.

*** Send emails to lecturer (gaye.stephens@tcd.ie) or TAs Seth (banagas@tcd.ie) or Tunji (omoniwab@tcd.ie) outside of any scheduled class times with any queries you have or clarifications that you need.**

Please let me know if you have any suggestions or spot any errors.

Part 2 eXtensible Markup Language - extract from the assignment sheet on blackboard.

1. XML and DTD documents

   a. From your group's UML Class diagram, pick <u>at least</u> 6 classes and for each create a different XML document. Include the following characteristics <u>for each</u> XML document:

      - At least 6 different XML elements/tags are used.
      - At least one third of the XML elements should have 1 XML attribute
      - Interlinks between some of the documents (reflecting the assocations/relationships between the classes within the UML design), with enough information to allow for interesting cross document XML Queries to be designed

   b. For each XML document create a DTD

2. Design and Document **at minimum 8 interesting XQuery** queries that support some of your UML use cases. **Pesent these queries during online sessions.**

   - At least 3 of the queries should retrieve information from two or more interlinked XML documents, using the WHERE clause
   - At least 2 of the queries should use the FOR clause
   - At least 1 of the queries should use the LET clause
   - At least 2 of the queries should use a Built-in XQuery function
   - At least 2 of the queries should use User Defined Functions

3. Present your XML, DTD and XQueries in a **group report** which also includes the following.

   - What (if anything) did you need to change in going from UML design to XML implementation? - Include revised diagrams/ethics canvas, if appropriate.
   - List who did what in the group for XML implementation
   - Strengths and Weaknesses of the XML design and XQueries design
   - For the XML and DTD documents- Use comments to clearly state what is the purpose of the document, and comments describing purpose of each element and for each attribute, and why certain cardinality (*,+ etc.) is used.
   - For each Xquery include: identification of the UML use case that it supports, description of the purpose of the query and provide example outputs that you expect when query is executed.

Past XML Exam Question

2.

(a) Explain using examples what constitute a well formed and valid XML document.

[10 Marks]

(b) Use DTD Notation to fully describe the XML document shown in Figure A above. Provide explanation for your design decisions

[16 Marks]

(c) Define and explain XQuery Statements for each of the following queries posed over the document in Figure A. Show expected results and explain your design decisions

I. Return within a single new element called "Colleagues", all the last name values in the document separated by a "+" sign.

II. Return just the values of the "medicalregnumber"attribute in a new element called "RegNumbers"

III. Return only the first of the firstname for each Doctor in the document

[24 Marks]

[Total 50 Marks]

Past exam papers can be found at
https://www.tcd.ie/academicregistry/exams/past-papers/annual/

XML

```
<?xml version='1.0' encoding='UTF-8' ?>
<DoctorDirectory>
<doctor area="Dublin" medicalregnumber="123456">
        <name>
                <firstname>James</firstname>
                <lastname>Murphy</lastname>
        </name>
        <telephone Type ="mobile">
                <number>0871234567</number>
        </telephone>
         <telephone Type ="Landline">
                <number>014587884</number>
        </telephone>
</doctor>
<doctor area="Kildare" medicalregnumber="789112">
        <name>
                <firstname>Freda</firstname>
                <firstname>Anne</firstname>
                <lastname>Hartigan</lastname>
        </name>
        <telephone Type ="mobile">
                <number>0879922345</number>
        </telephone>
         <telephone Type ="Landline">
                <number>045865768</number>
        </telephone>
</doctor>
<doctor  medicalregnumber="223445">
        <name>
                <firstname>Francis</firstname>
                <firstname>Mary</firstname>
                <lastname>Kelly</lastname>
        </name>
         <telephone Type ="Landline">
                <number>04487994</number>
        </telephone>
</doctor>
</DoctorDirectory>
```

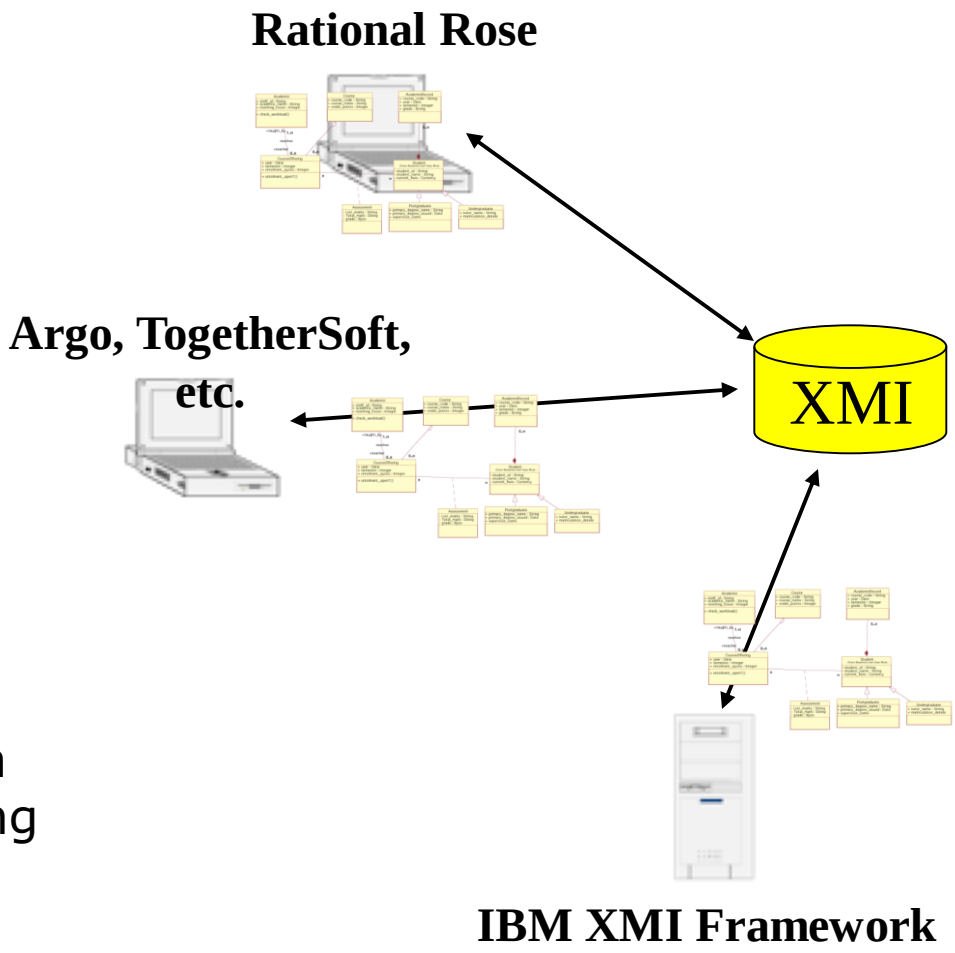
Figure A for exam question on previous slide

Generating XML Documents from UML Instances

Check out the short video by Seth Banaga on Blackboard showing an example.

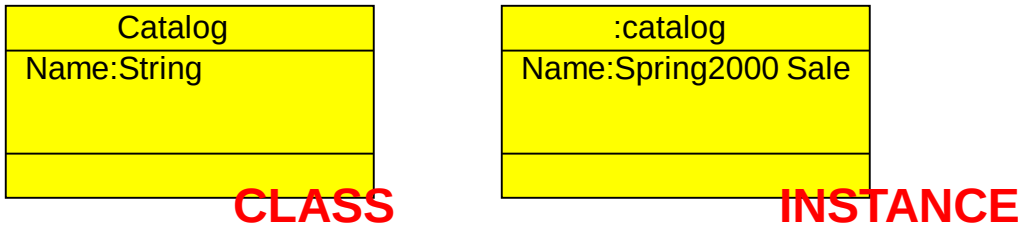
XML MetaData Interchange: XMI

- Standard sponsored by the OMG
- Originally for allowing interchange of UML models between UML editors
- Now seen as sensible XML representation of UML for other purposes
  - E.g. XML representation of entities specified using UML



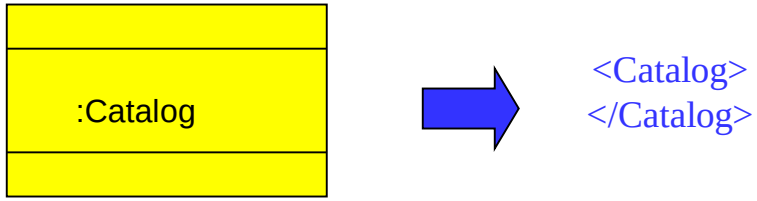
Moving from UML to XML

- Want to generate
  - XML document instance from UML instance model



UML Class mapping

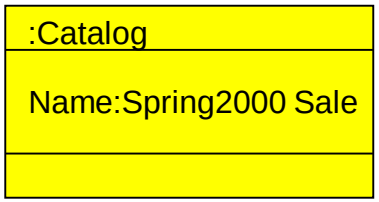
- Each instance of a UML class produces one XML element



- UML class name translates to XML tag name
- Be careful in naming your UML classes as XML has restrictions on valid tag names
  - Cannot contain spaces
  - Alpha or Number characters but also full stop, dash or underscore (. - _)
  - Can begin with letter or _
  - CANNOT begin with letters XML !!

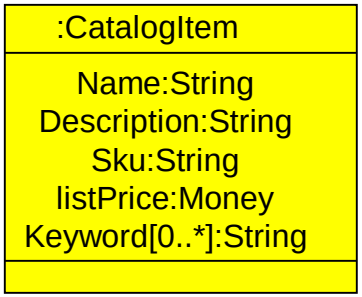
UML Attribute mapping

- Each attribute of a UML class produces a child XML element
- Element name is made unique by prepending with the class name




```
<Catalog>
  <Catalog.name>
    Spring 2000 Sale
  </Catalog.name>
</Catalog>
```

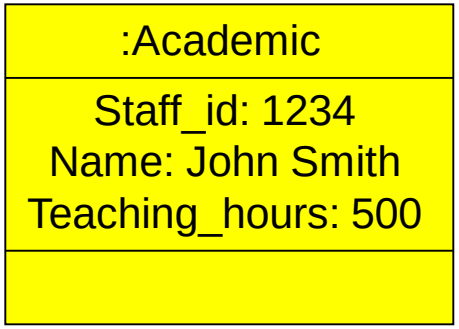
- XML has no representation for multivalued attributes of UML so these attributes are translated into individual XML elements
  - E.g. keyword[0..*]:String



values for keyword attributes in Instances of CatalogItem are represented in XML like this

```
<Catalogitem.keyword> Personal Computer </Catalogitem.keyword>
<Catalogitem.keyword> Windows 2000 </Catalogitem.keyword>
<Catalogitem.keyword> Notebook </Catalogitem.keyword>
```

Another Example




```
<Academic>
   <Academic.staff_id> 1234 </Academic.staff_id>
   <Academic.name> John Smith </Academic.name>
   <Academic.teaching_hours> 500 </Academic.teaching_hours>
</Academic>
```

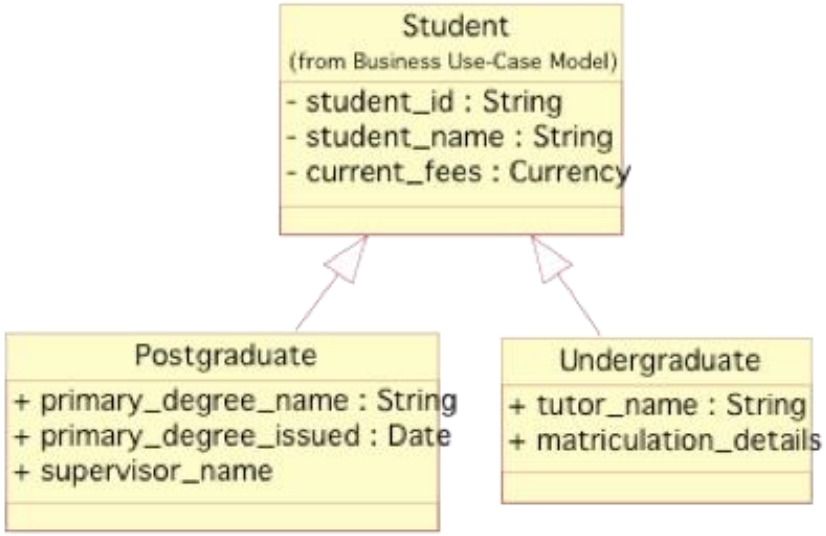
UML Inheritance mapping

- Current XML standards do not have built in mechanism for representation of inheritance

- The 'XMI standard' specifies use of "copy down" approach for generalisations, attributes, association refs and compositions

  - That is definitions from all superclasses are copied down to the class being translated into XML

An example of mapping Inheritance

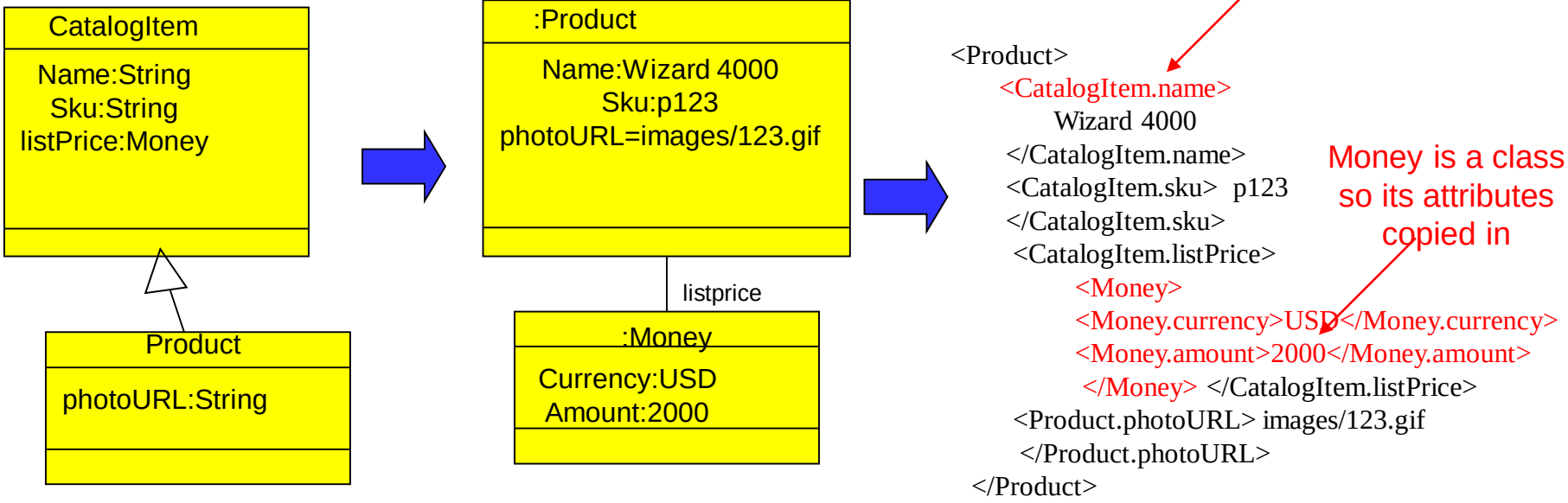


:Postgraduate

Student_id: 99124
Student_name: Frank Clarke
Current_fees: 6500
Primary_degree_name:BA
Primary_degree_issued:12 Nov 2003
Supervisor_name: John Smith

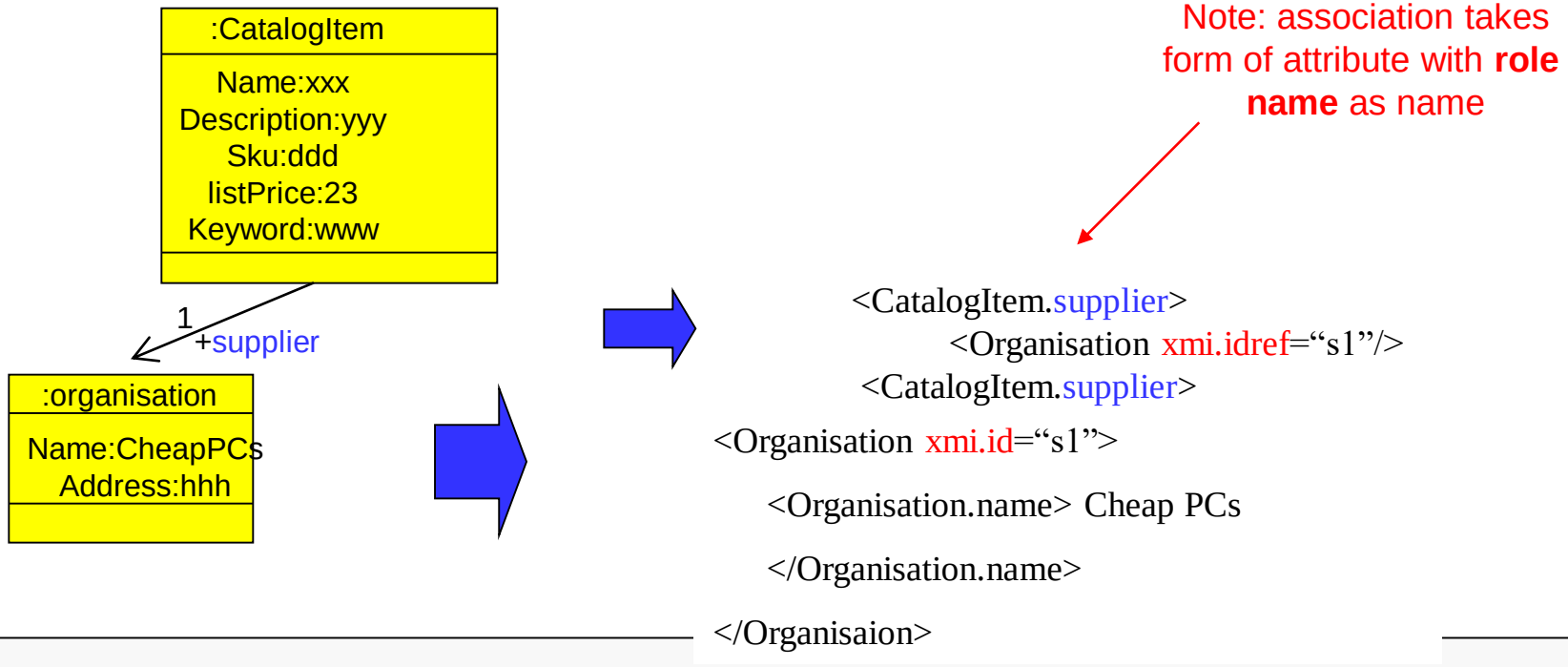
```
<Postgraduate>
   <Student.student_id> 99124 </Student.student_id>
   <Student.student_name> Frank Clarke </Student.student_name>
   <Student.current_fees> 6500 </Student.current_fees>
   <Postgraduate.primary_degree_name> BA </Postgraduate.primary_degree_name>
   <Postgraduate.primary_degree_issued > 12 November 2003 </Postgraduate.primary_degree_issued>
   <Postgraduate.supervisor_name > John Smith </Postgraduate.supervisor_name>
</Postgraduate>
```

UML Inheritance mapping

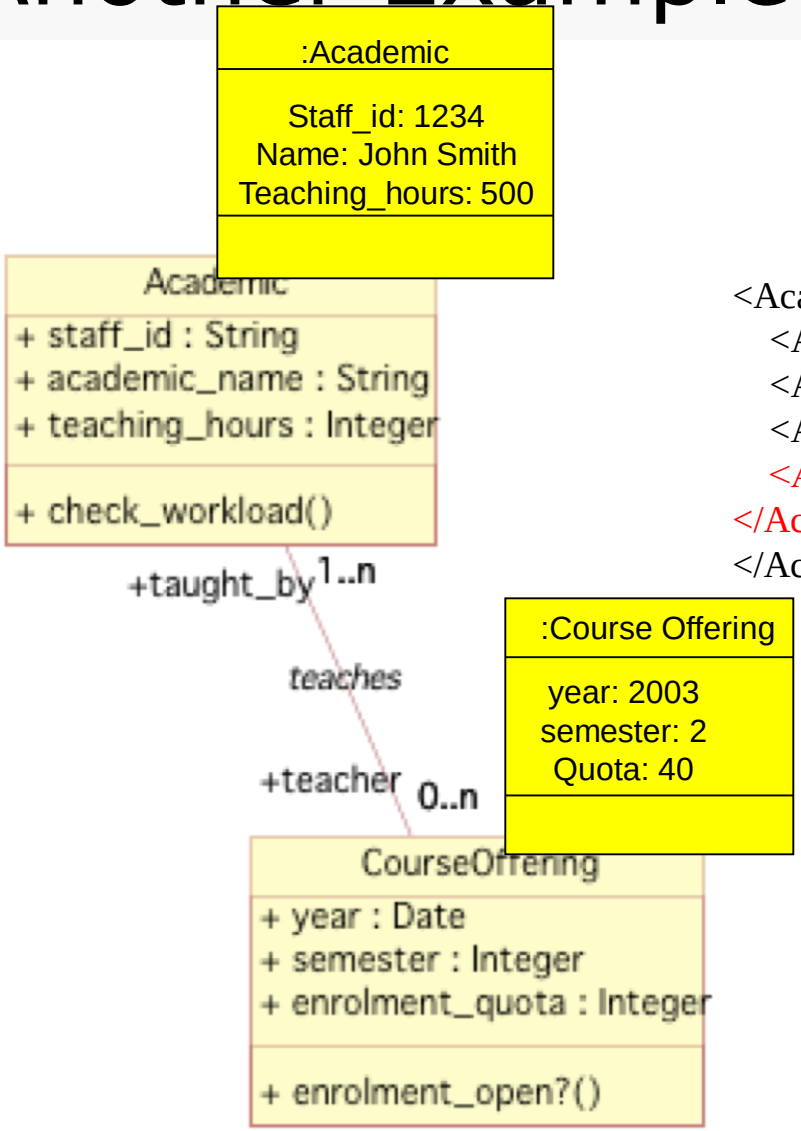


UML Associations
Simple approach

- A reference to the class of the associated class is included in the definition using the xmi.idref attribute
  - xmi.id then used to label definition of class



Another Example




```
<Academic xmi.id="22">
   <Academic.staff_id> 1234 </Academic.staff_id>
   <Academic.name> John Smith </Academic.name>
   <Academic.teaching_hours> 500 </Academic.teaching_hours>
   <Academic.teacher> <CourseOffering xmi.idref="4ba5"/>
</Academic.teacher>
</Academic>
```

```
<CourseOffering xmi.id="4ba5">
   <CourseOffering.year> 2003 </CourseOffering.year>
   <CourseOffering.semester> 2 </CourseOffering.semester>
   <CourseOffering.enrolment_quota> 40 </CourseOffering.enrolment_quota>
   <CourseOffering.taught_by> <Academic xmi.idref="22"/>
</CourseOffering.taught_by>
</CourseOffering>
```

BaseX Software

- A light-weight, high-performance and scalable **XML Database** engine and **XPath/XQuery Processor.**

- Interactive and user-friendly **GUI frontend**

- Different programming APIs to connect to BaseX XML database
  - REST-Style Web API
  - Variety of Client APIs for different programming languages See http://docs.basex.org/wiki/Developing

- YOUR ACTION: Download Core Package Java BaseX to your laptop or your U: drive or to D: drive on PC (http://basex.org/products/download/all-downloads)

That's All Folks Thank You for Listening

How Many Programmers does it take to change a lightbulb?

None, it is a hardware problem.