

UML

Unified Modeling Language

COMP 3111/H tutorial

Class Diagram in Unified Modeling Language (UML)

- A UML class diagram describes the structure of a system by showing the classes, their attributes, operations (or methods), and the relationships (i.e. associations) among objects
- Once the UML class diagram is modelled, it helps translate to the corresponding programming codes

UML Class Diagram

- Entities
- Attributes
- Operations (functions, methods...)
- Associations and multiplicities
- Building a Class Diagram using ArgoUML

Design a Class Diagram

Storyline

- A university hosts a number of **faculties** (i.e. **schools in HKUST**)
- Every faculty contains a number of **departments**
- Each department offers a number of **courses**
- The courses may be offered in different **semesters**
- Each course will be taught by **instructor(s)** and **students** can register for it through the **Registrar office** (i.e. **ARRO in HKUST**).

Design a Class Diagram

Knowing the Entities

- Faculty
- Department
- Course
- Semester
- Instructor
- Student
- Registrar Office

Design a Class Diagram

Defining the attributes

- The attributes are the properties of each class
- Examples:
 - Course {code, credit, meeting hours, ...}
 - Student {full name, birth date, national id, ...}
- What about Faculty, Department, Semester, Instructor, Registrar Office?

Design a Class Diagram

Defining the functions (operations)

- The functions represent the actions that can be done by entities
- Examples:
 - A **student** can register a **course**
 - a **Registrar Office** can
 - open registration
 - close registration and
 - accept **course** requests
- Each function can take multiple input parameters and return only one object

Design a Class Diagram

Defining the relations between classes
(Associations)

- The **Faculty** has **more than one Department**, and the **Department** has **more than one Course**
- How to define the above relationship in UML Class Diagram?
 - Draw an association link (i.e. a straight line)
 - Draw the name of relation (optional)
 - Add multiplicity factors on both ends
 - Examples: 1...*, *, 0...5

Example: Defining an association

- The **Faculty** has more than one **Department**, and the **Department** has more than one **Course**



Try to read along this direction

What should I do if the requirements are not clear?

- If the requirements are not clear, the multiplicities can be deduced by common senses
- Example:
 - Department should only be assigned to one Faculty (i.e. school)
 - Each course can only be offered by one Department



What happens if the requirements are change?

- For example, there is an Interdisciplinary Programs Office (IPO) and it may better be managed by at most 2 faculties (i.e. Schools)?
- The UML diagram (and later the programming logic) should also be changed accordingly



Design a Class Diagram

Sample Designs

Table 1: Sample Design for the University System Classes

Class Name	Attributes	Operations
Faculty	Name : String	void AddDepartement (dep : Departement)
Department	Name : String	void AddCourse (course : Course)
Course	Code : String CreditHours : Integer	
Semester	Year : Integer Season : String	void RegisterCourse (course : Course , instructor : Instructor)
Instructor	Name : String BirthDate : Date Room : String	
Student	Name : String BirthDate : Date RegistrationYear : Integer	
RegistrarOffice		void OpenRegistration (void) void CloseRegistration (void) void RegisterCourse (student : Student , course : Course , semester : Semester)

Design a Class Diagram

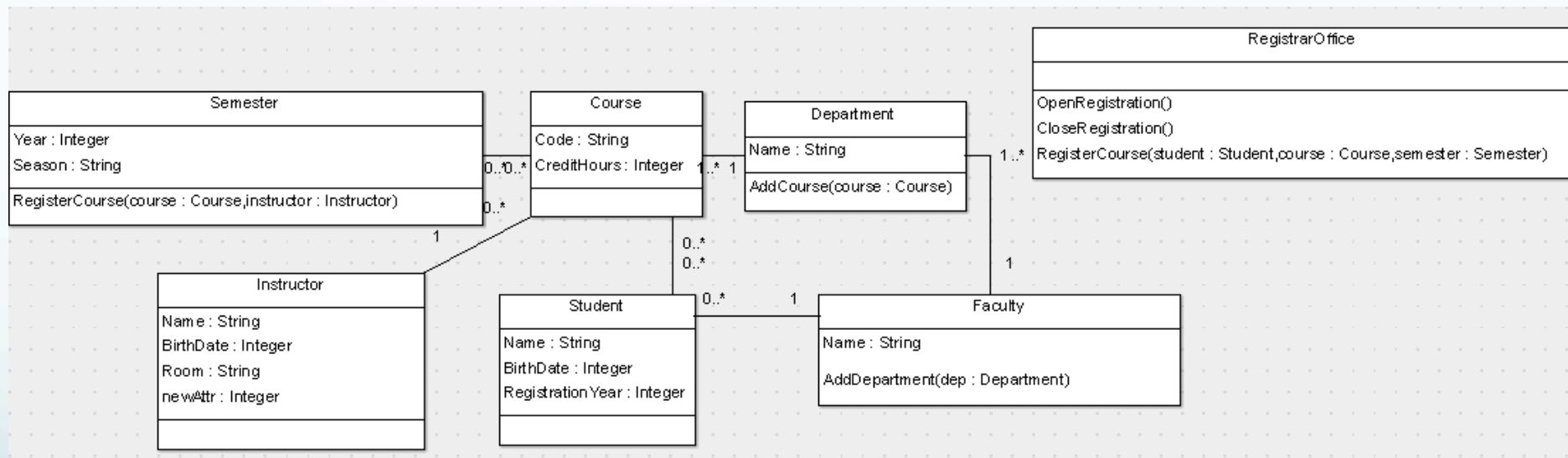
Sample Designs

Table 2: Sample Associations between classes

Right Side Class	Multiplicity	Multiplicity	Left Side Class
Faculty	1	1..*	Department
Faculty	1	0..*	Student
Department	1	1..*	Course
Course	0..*	0..*	Semester
Course	0..*	1	Instructor
Course	0..*	0..*	Student

Design a Class Diagram

Sample Designs



UML Use Case Diagram

- Actors
- Use cases

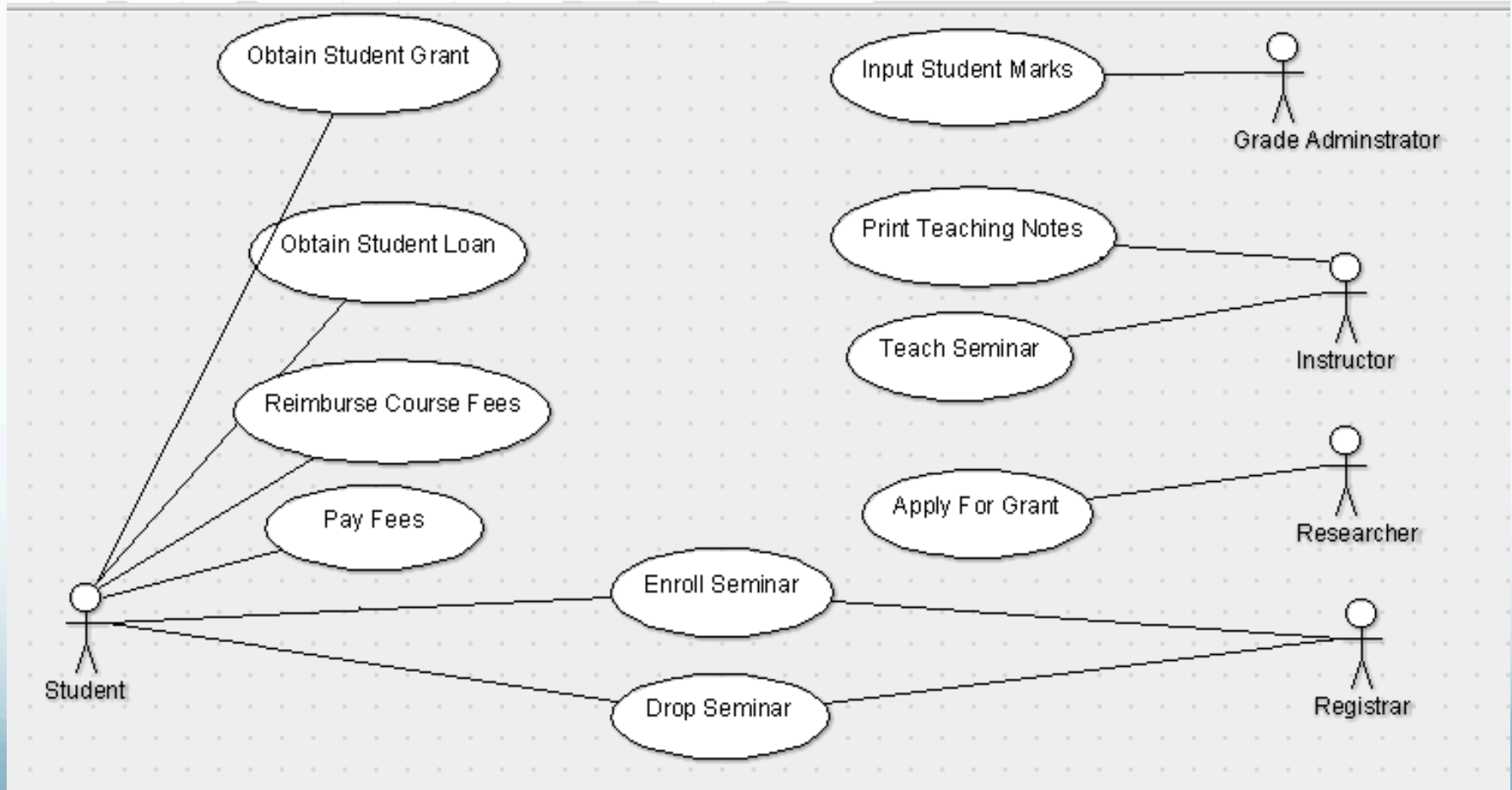
Building the Use Case Diagram

(Add Use Cases)

- Student (as an actor) can complete lots of things using the system
 - Pay tuition fees
 - Reimburse course fees
 - Obtain grant / loan
 - Enroll / Drop seminar

Building the Use Case Diagram

(Sample use case diagram)



What is ArgoUML?

- It is a software to draw UML diagrams
- ArgoUML is the leading open source UML modeling tool and includes support for all standard UML 1.4 diagrams
- URL:
 - <http://argouml.tigris.org/>



Start ArgoUML Now (two methods)

1. Download and install:

Download and install ArgoUML, using the link above.

2. Launch via Java Web Start:

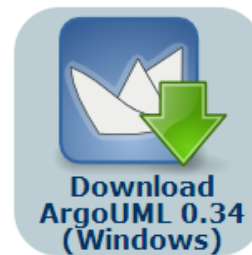
[Click](#) to launch the latest stable release.

If this doesn't work, try updating your Java version at java.com or reading the [Java Web Start product page](#). If you have problems, please see the [FAQ](#).

[Other formats](#)

Startup ArgoUML

- 3 methods
 - Download and install ArgoUML
 - Launch it via Java Web Start
 - Run the binary packages
 - Click other formats
 - Download the ArgoUML.zip
 - Unzip
 - Double-click the argouml.jar



Start ArgoUML Now (two methods)

1. Download and install:

Download and install ArgoUML, using the link above.

2. Launch via Java Web Start:

[Click](#) to launch the latest stable release.

If this doesn't work, try updating your Java version at java.com or reading the [Java Web Start product page](#). If you have problems, please see the [FAQ](#).

[Other formats](#)

Main Screen

