

Tutorial 2

Today we will have a quick introduction to ArgoUML and how to use it, as well as look at some examples of class diagrams.

This is an answer sheet, answers are marked in RED.

Quiz dates:

- **Quiz 1**, available from 9am Friday 13 Mar to 11:30pm Monday 16 Mar.
- **Quiz 2**, available from 9am Friday 3 Apr to 11:30pm Monday 6 Apr
- **Quiz 3**, available from 9am Friday 8 May to 11:30pm Monday 11 May
- **Quiz 4**, available from 9am Friday 29 May to 11:30pm **Tuesday** 2 Jun. Note: Monday 1 June is the Queen's Birthday holiday.

Review

Last week we looked at how to use Eclipse and an introduction to using Java. Try answering the following question about the Java `main()` method.

Q1: What do each of the keywords of the following method signature mean?

```
public static void main(String[] args) { ... }
```

public	This method is accessible to the Class, Package, Subclass and anywhere this package is used (AKA "World").
static	This member belongs to the <u>class</u> and not the <u>instance</u> (object). It means that only one instance of a static field exists even if you create a million instances of the class or you don't create any. It will be shared by all instances.
void	This method does not return any value.
String[] args	An array of String objects representing the command-line arguments this application received when it was run.

Tutor Contact Details

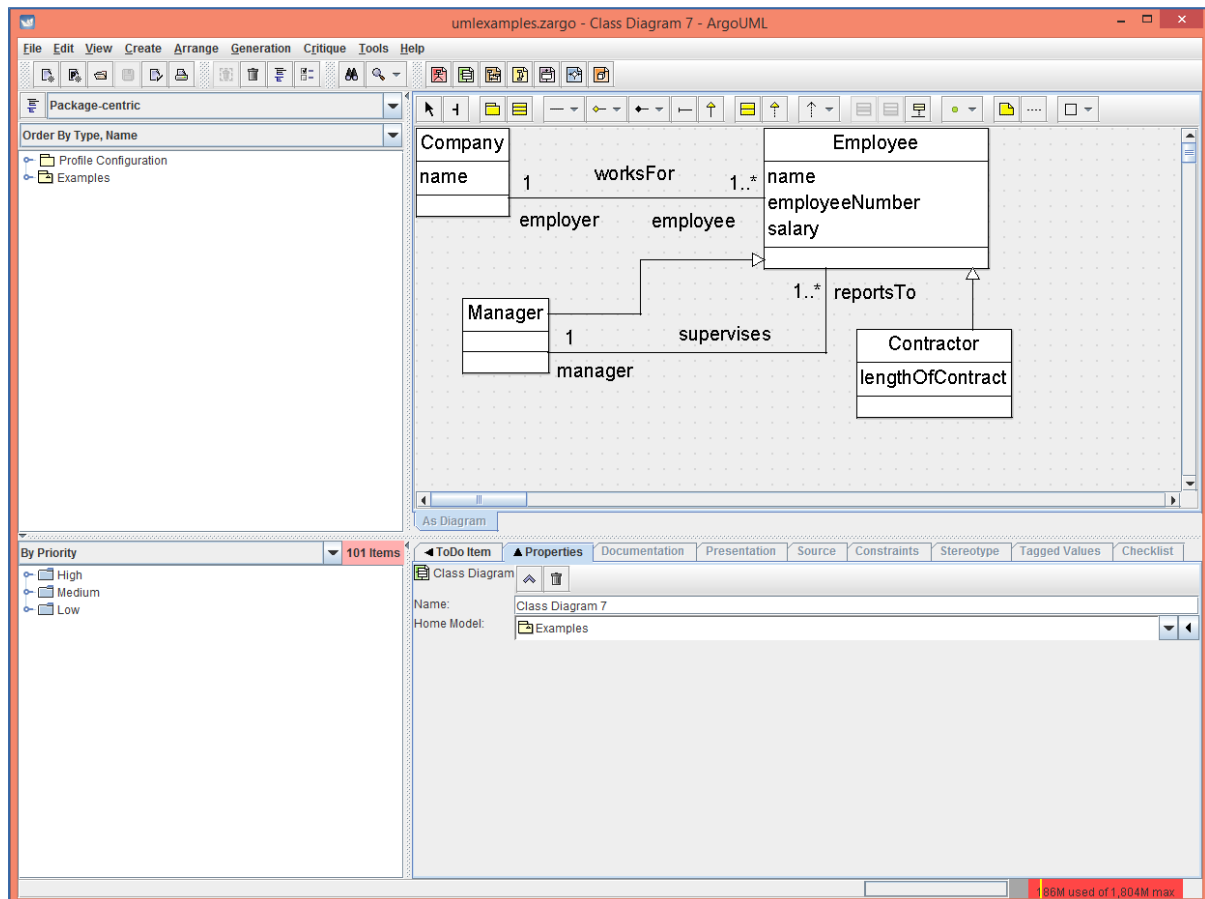
- **Freddy Hutchinson:** fhut994@aucklanduni.ac.nz
- **Monica Bian:** rbia002@aucklanduni.ac.nz

Part A – Installing & using ArgoUML

ArgoUML is a diagramming tool designed for creating and modifying UML diagrams. Its use is required by your first assignment in CS230.

- **Step 1:** Go to <http://argouml-downloads.tigris.org/argouml-0.34/>
- **Step 2:** Download ArgoUML-0.34.zip - **Binary distribution (.zip format).**
- **Step 3:** Save it somewhere on your H:\ drive (or, alternatively your home computer or USB memory drive)
- **Step 4:** Unzip the .zip file.
- **Step 5:** Inside the extracted directory, open argouml.jar
 - ArgoUML should now launch
- **Step 6:** Download umlexamples.zargo from the course assignment page and save it as above
- **Step 7:** (In ArgoUML) use File > Open Project to open the above .zargo file.

You should now see something like this:



Aggregation implies a relationship where the child can exist independently of the parent. Example: Class (parent) and Student (child). Delete the Class and the Students still exist.

Composition implies a relationship where the child cannot exist independent of the parent. Example: House (parent) and Room (child). Rooms don't exist separate to a House.

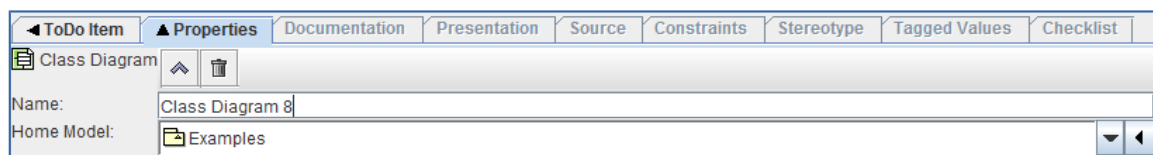
The above two are forms of **containment** (hence the parent-child relationships).

Dependency is a weaker form of relationship and in code terms indicates that a class uses another by parameter or return type.

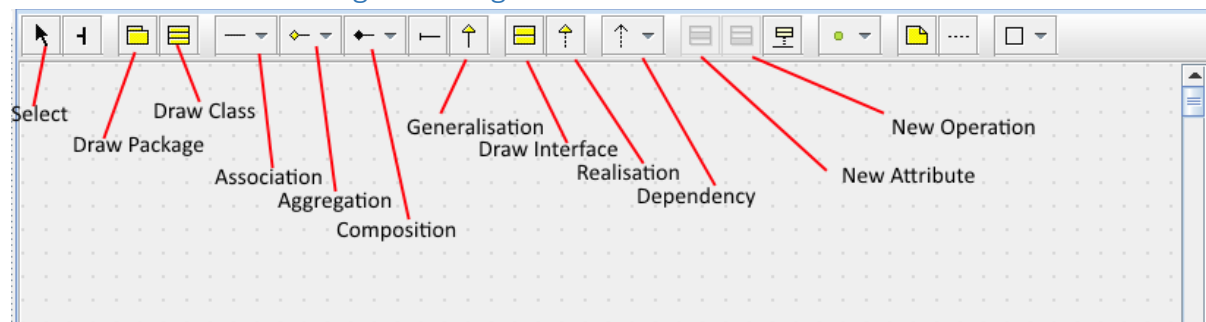
Dependency is a form of **association**.

Part B - Class Diagrams

- **Step 1:** In ArgoUML go to Create > New Class Diagram
 - You should be presented with a blank canvas.
- **Optional Step 2:** In the properties pane, you can rename your diagram by modifying the Name field illustrated below.



How to use the Class Diagram Designer



Task 1:

Now that you have ArgoUML installed and a brief overview of its features, you now have the opportunity to practice. Use the above tools to draw the following Class Diagram.

Outline:

A local pet store has contracted you to design a system to keep track of their stock, orders and customers. Before you start work on any code, they want to see what your design looks like. Draw a simple UML Class Diagram that conveys the following basic concepts.

- An "Order" class to keep track of each order a customer makes. It needs to have (at minimum) the properties of "date", "price", "customer", and "animals" that were sold.

- An “Item” class that has a properties “type”, “price” and “quantity” for items like Toys, birdcages and fishtanks.
- An “Animal” class that extends (or *generalises*) “Item”, to keep track of each animal in their stock. An animal needs the further properties of “breed” and “isFed”. It also needs the member (operation) of “Feed()”
- A “Customer” class to keep track of each customer, with the properties “name” and “address”.

Think about the *types* the following variables would need to have if they were implemented as a class. The following are some type suggestions: float, boolean, String, Date.

Solution:

Part C – Use-Case Diagrams

- **Step 1:** using ArgoUML go to Create > New Use Case Diagram

Task 2:

For the example described above in part B, draw a use-case diagram to represent the following cases:

- A customer examining/petting an animal.
- A customer purchasing an animal.
- A staff member feeding the animals.

Solution: