COMP1531

Week 4 Tutorial!

Housekeeping

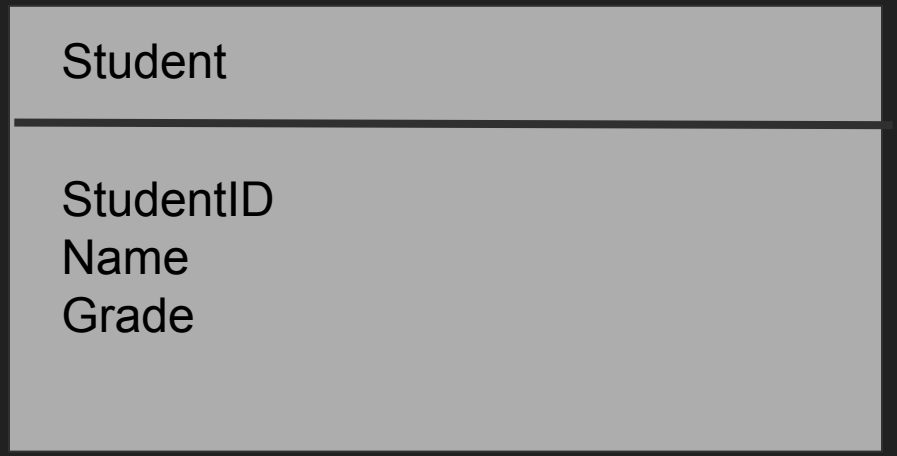
- Project deliverable 1 marking
- Mid-term exam (**individual task**):
  - Week 5 Friday midday
  - Take-home (open book)
  - No more than 1 hour (with prep beforehand)
  - 10 MCQ (wk1-4), 1 design question
- Lab 3 due this Sunday
- No lab this week
- Quiz 1 due Mon 18th

Fundamental design principles of OO

I. Classes and Objects

Class

- A class is a blueprint for many objects that share common properties



Object

- An object is an instance of a class
- Each unique student is an instance of the class Student

Another example:

- class Circle:
  - radius
  - colour
- a red circle with radius 4cm is an instance of the class

II. Abstraction, Inheritance and Association

Concept of Abstraction

- More abstraction means less detailed

- Imagine if you’re short-sighted, you walk along the street and you see two blobs - high level of abstraction
- Then you found your glasses and you look at the them again: hey! it’s a cat and a dog! - lower level of abstraction, more details

Abstraction in Programming

- High level: It’s 7am, you want coffee, you go to the coffee machine and press a button. Voila! Coffee! You don’t care how it works inside.
- Low level: The mechanics inside the coffee machine

In programming:

- High level: using a function (black box)
- Low level: looking at function implementation (white box)

Abstraction in OO

Class:

- objects
- methods that modify attributes of objects
- can’t access any other way (Encapsulation)

Encapsulation

- Encapsulation of behaviour implies that clients are abstracted from the internal implementation of a method and only interact with the method through its defined signature.
- Bank account balance example

Inheritance

What if we have more than circles?

Class Circle:
- radius
- colour

Class Square:
- side length
- colour

Grouping together

Abstract Class Shapes:
- Colour

Class Circle(Shapes):
- Colour
- Radius

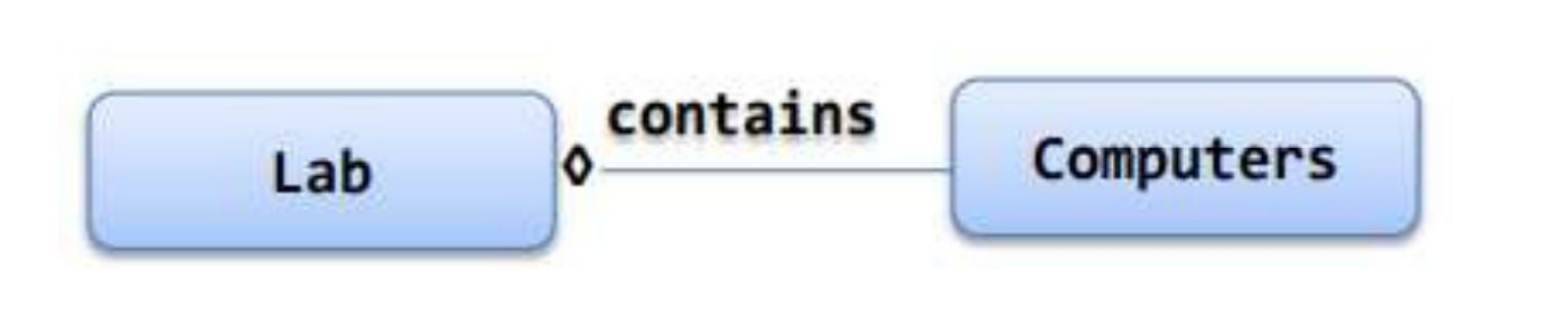
Class Square(Shapes):
- Colour
- Side-length

Types of Associations

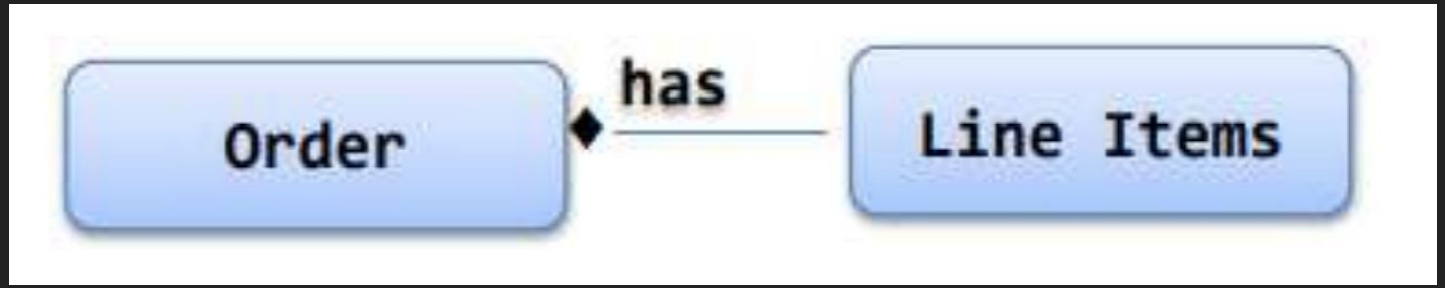
3 types

- Aggregation
- Composition
- Association (when the above don’t apply)

Aggregation



Composition



Use-case analysis vs domain-modelling

- Use-case analysis (Black box)
- Domain-Modelling (Transparent box)

Example of Domain Models

- CRC cards
- UML Diagrams

CRC

- Class-Responsibilities-Collaborations
- Helps evolve into UML Diagram

- attributes (knowledge the object has) - Top
- methods (actions the object can perform) - Left
- collaborators (other classes interacts with this one) - Right

Let’s Practice!

- Restaurant
- Administrators
- Admin system
- Tables
- Orders

Converting to UML Diagram

- board-work