

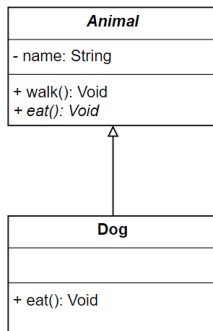
## COMP2511 Lab03 UML Diagram Notation

### Basics

- Make sure your UML diagrams are all legible - e.g. not split across multiple pages

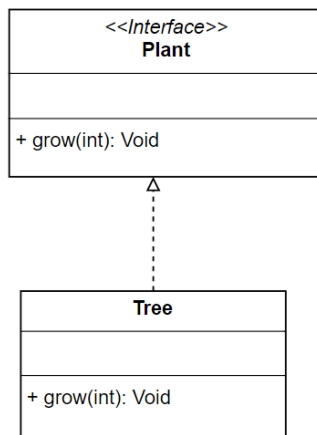
### Abstract Classes and Interfaces

**Abstract classes** should be *italicised*. Abstract methods should also be italicised. All implementations of abstract methods should be explicitly listed out.



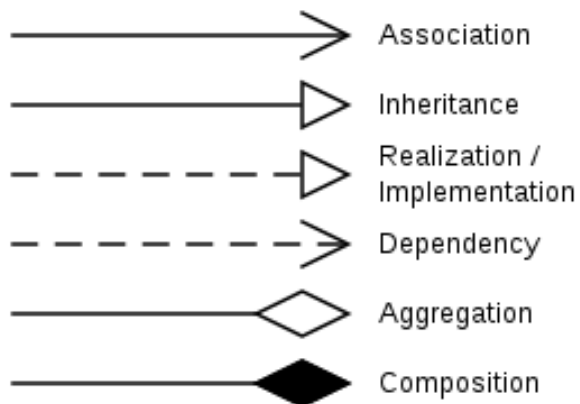
The abstract class *Animal* has the abstract method *eat* and the regular method *walk*.

**Interfaces** should have `<<Interface>>` preceding the name. All implementations of interface methods should be explicitly listed out.



The interface Plant has the method `grow(int)`.

## Relationships



Association - a class uses another in some way.

Inheritance - a class inherits another. The arrow points to the parent class.

Implementation - a class implements an interface. The arrow points to the interface.

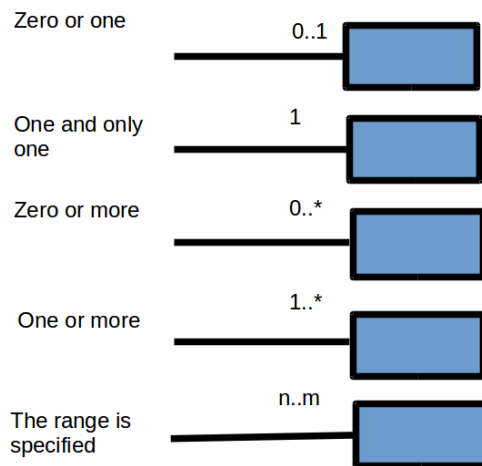
Dependency - a class depends on another.

Aggregation ("has-a" relationship) - a class "A" contains another class "B". "B" **can** exist independently of "A". The diamond is on the side of "A" (the container)

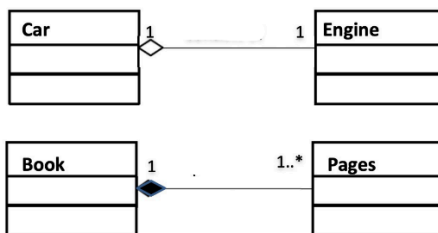
Composition ("has-a" relationship) - a class "A" contains another class "B". "B" **cannot** exist independently of "A". The diamond is on the side of "A" (the container)

## Cardinality

All has-a relationships **must have cardinality**. Some common examples are shown below.



Cardinality must be depicted on both sides of the relationship. In these examples, 1 car has 1 engine, while 1 book has 1 or more pages.



## Access Modifiers

- + denotes public fields and methods
- - denotes private fields and methods
- # denotes protected fields and methods
- Static fields and methods are underlined

In this example, Animals have the private field name and the public methods walk() and eat().

