UML Diagram Cheatsheet

Tips and Resources

Make sure your UML diagrams are all on one page. Some sites slice the UML diagram up into separate pages when you export it as a pdf, which makes it very difficult to mark!

`Although the specs usually ask for a .pdf format, as long as it is legible and in your repo it is okay as they are manually marked.`

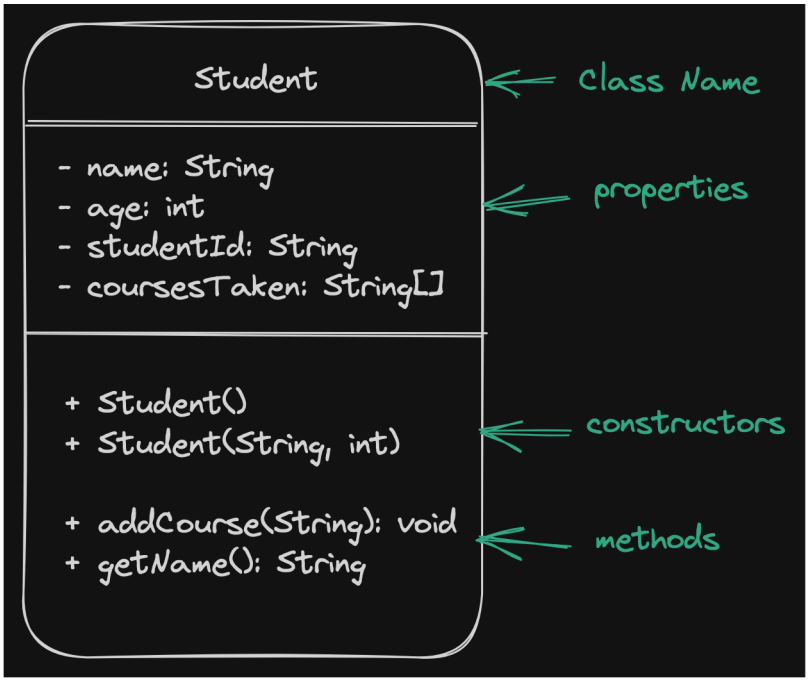
Don't forget to add cardinalities. If a class has a composite relationship (Aggregation / Composition), you must state the cardinalities (ie. 1..0*).

`You will be marked on this on your assignments and exam.`

Where to create diagrams

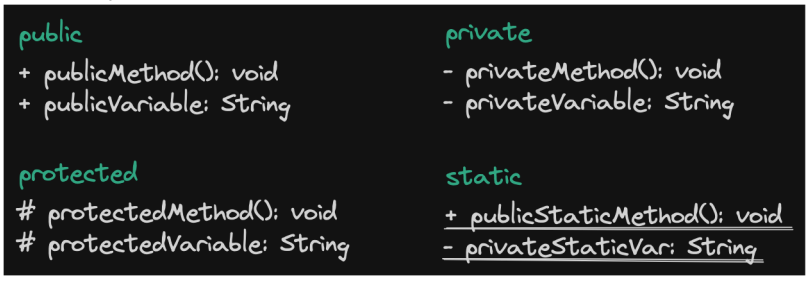
- draw.io: https://app.diagrams.net/
- lucid: https://lucid.app/

Basic Class



Access Modifiers

Modifier syntax applies for both attributes and methods



Abstract Classes and Interfaces

Abstract Classes

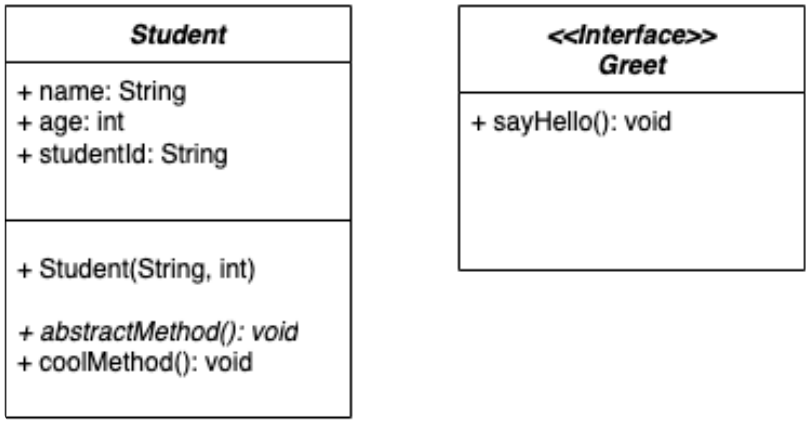
Abstract classes should be *italicised*

- Abstract methods should also be italicised
- Abstract and implemented methods should all be listed

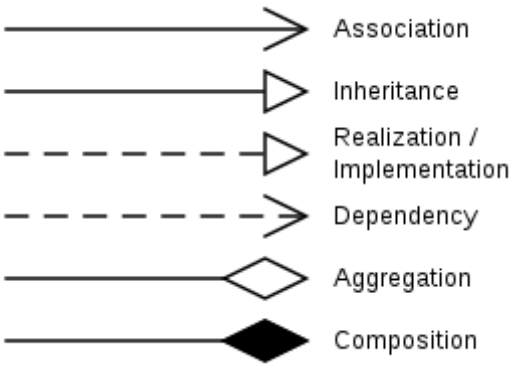
Interfaces

Interfaces should have <<*Interface*>> preceding the name

- Although interface methods are abstract by nature, we won't penalise you for not italicising the method stubs



Relationships



Association - a class uses another in some way. Uncommon in this course.
Inheritance - a class inherits another class. Arrow towards parent class
Implementation - a class implements an interface. Arrow towards interface
Dependency - a class depends on another. Uncommon in this course.
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 classes that have a "has-a" (composite) relationship must have their cardinalities written on both classes (near the head/tail of the arrow).