Basic UML Class Diagram Notation

Class

Name
attributes (member variables)
methods (member functions) + public_method() # protected_method() - private_method()

Abstract class

Name
virtual method() method()

Name {Abstract}

Inheritance (is-a) relationship

Base

Derived2 is-a Base

Derived1   Derived2

Object

classname: objectname
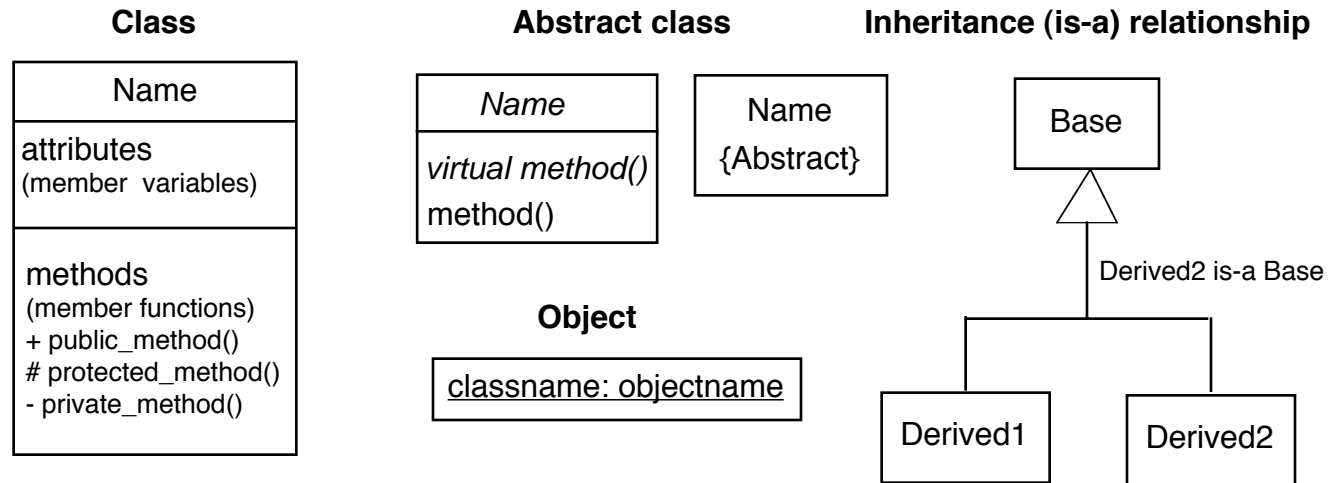
Aggregation and Composition (has-a) relationship

Whole

◇

Part

Whole has Part as a part;
lifetimes might be different;
Part might be shared with other
Wholes.
(aggregation)

Whole

◆

Part

Whole has Part as a part;
lifetime of Part controlled by Whole,
Part objects are contained in one
Whole object.
(composition)
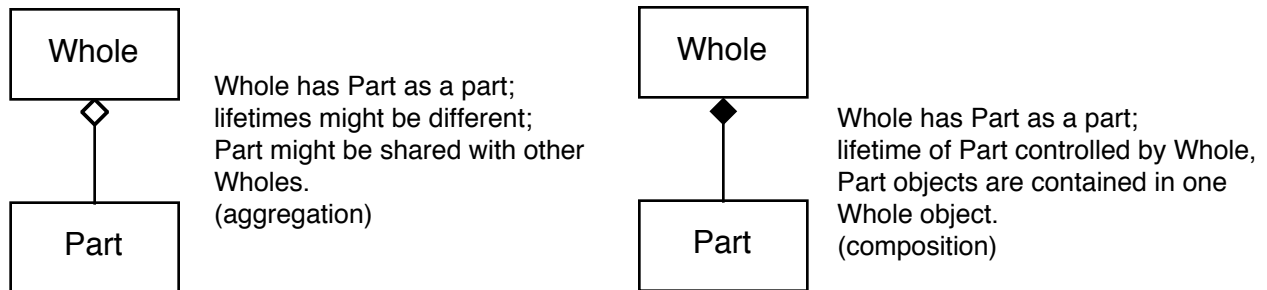
Association (uses, interacts-with) relationship

A —— B

A's role   B's role

Navigability - can reach
B starting from A

A ——▶ B
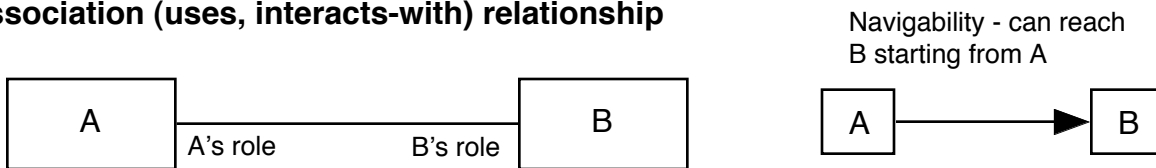
Multiplicity in Aggregation, Composition, or Association

* - any number	0..1 - zero or one	Follow line from start class to end class,
1 - exactly 1	1..* - 1 or more	note the multiplicity at the end.
n - exactly *n*	*n .. m* - *n* through *m*	Say "Each <start> is associated with <multiplicity> <ends>"

A   1   *   B
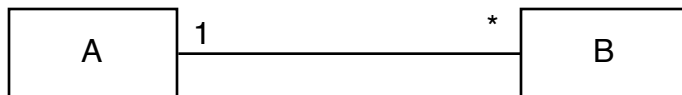
Each A is associated with any number of B's.
Each B is associated with exactly one A.

Basic UML Sequence Diagram Notation

objects

one that starts the action at the left

object1

time flow

<<create>>

object2

get_info(spec)

search-self

requested info

do_something()

void return is implicit

<<destroy>>

---

object1

lifetime

focus of control

message sent
(function call)

information returned
(non-void return)

object sends message to itself
(one method calls another)

<<create>>

<<destroy>>

return with no information
(void return) - use for clarity only