J02 Introductory Java 2

Types
Objects
Classes
Interfaces
Inheritance

ANU – School of Computing – Structured Programming 1110 / 1140 / 6710

Objects

Objects combine state and behaviour

- **State**: fields (data)
- **Behaviour**: methods (code)
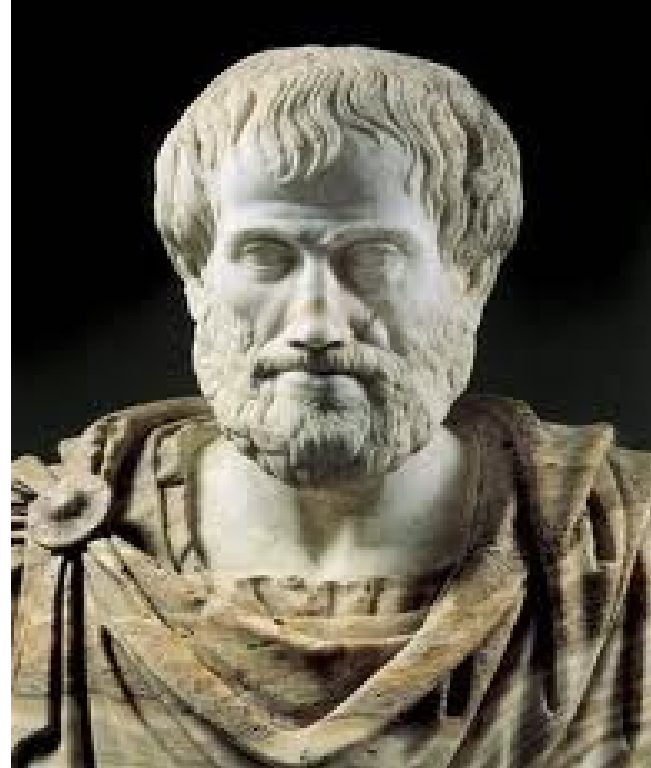
Example: Bicycle

- **State**: current direction, speed, cadence, gear
- **Behaviour**: turn, change cadence, brake, change gear

Classes

Aristotle 384-322 BC

"Blood-bearing animals'":

- Four-footed animals with live young,

- Birds,

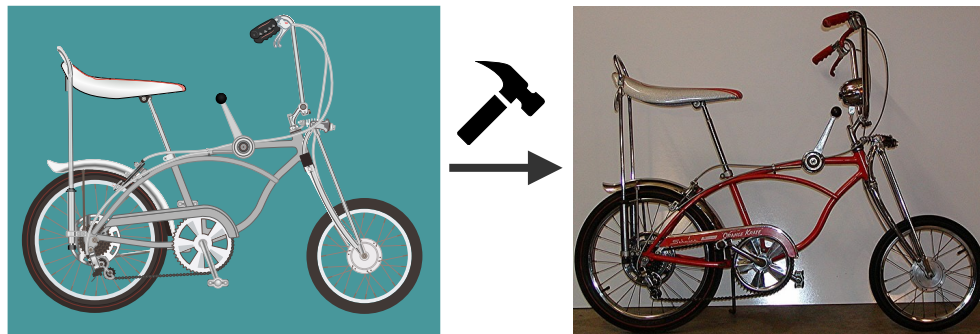- Egg-laying four-footed animals,

- Whales,

- Fishes

Classes

A class is a blueprint or 'type' for an object

- **Class**: blueprint / definition used for multiple instances
- **Instance**: one instantiation of a class (aka object)

Example:

- **Class**: Schwinn Sting-Ray
- **Instance**: your bike

Java Interfaces

An `interface` is a group of methods without implementations (methods define behaviour)

Example: an `interface` `MovableThing` might include:

- `brake()`

- `speedup()`

Any class that *implements* `MovableThing` must include definitions of these methods.

Inheritance

Classes may form a hierarchy

- sub-class `extends` a super-class

- child-class `extends` a parent-class

Example:

`SchwinnStingRay` `extends`

`WheelieBike` `extends`

`UprightBike` `extends`

`Bike` `extends`

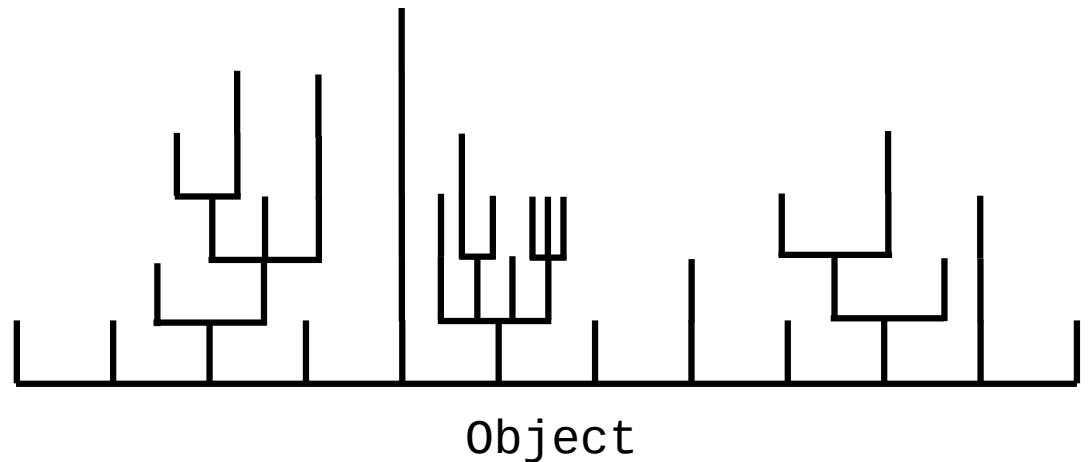`Object` (Root of all Java Classes)

Object

Image Attributions

- By Manotechnologie - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=48730292

- By Nels P Olsen - https://www.flickr.com/photos/7776581@N04/2011509547/in/photostream/, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=17156113

- By Devin McElheran - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=56317090