Australian
National
University

Introductory Java 2

J2

Types

Objects

Classes

Inheritance

Interfaces

Objects

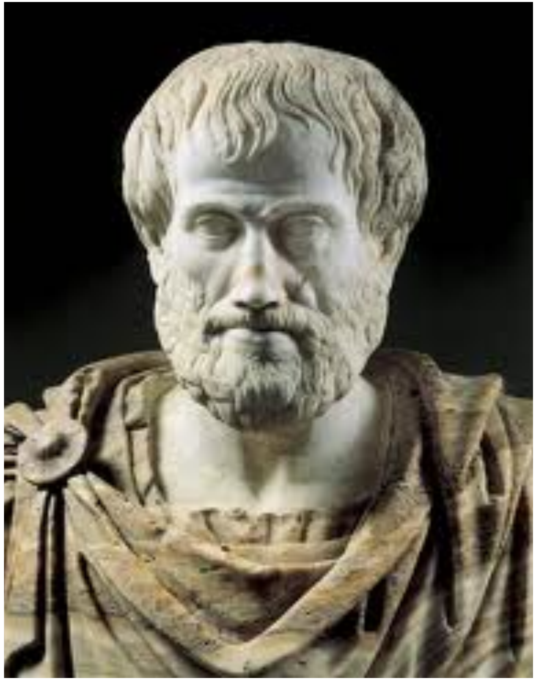
Objects combine state and behavior

- State: *fields* (data types)
- Behavior: *methods* (imperative code)

Example: bicycle

- State: current speed, direction, cadence, & gear
- Behavior: change cadence, change gear, brake

Classes



Aristotle 384-322BC

‘Blood-bearing animals’:

1. Four-footed animals with live young,
2. Birds,
3. Egg-laying four-footed animals,
4. Whales,
5. Fishes

Classes

A class is a blueprint or ‘type’ for an object

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

Example: bicycle

- Instance: your bike
- Class: Kona Jake The Snake 2012

Inheritance

Classes may form a hierarchy

- sub-class: `extends` a super-class

Example: bicycle

- class: `KonaJakeTheSnake2012`
- super-class: `CyclocrossBike`
- super-class: `UprightBike`
- super-class: `Bike`
- super-class: `Object`

Java Interfaces

Methods define behavior

- An `interface` is a group of methods *without* implementations

Example: an interface `MovableThing` might include:

- `brake()`
- `speedup()`

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