

Basics of Inheritance



Chapter 5 – Section 1



Table of contents

- ☉ Inheritance: the "IS A" relationship
- ☉ The special case of constructors
- ☉ The three roots of Inheritance



Reminder: the Object concept

the Object concept

Class name



Attributes



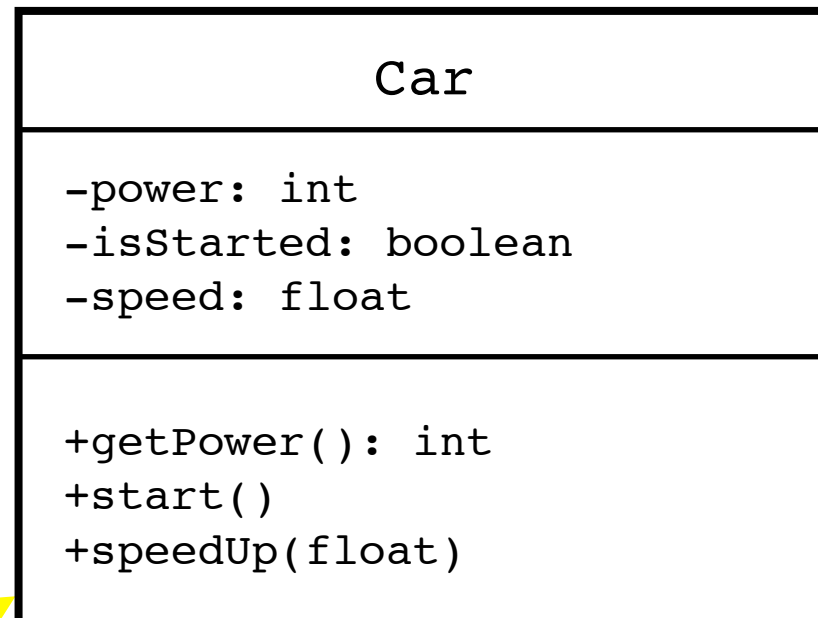
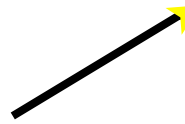
Methods



Accessors

Modifiers

Visibility





Reminder: the Object concept

Attributes →

Accessors →

Modifiers →

Visibility ↗

```
public class Car {  
    private int power;  
    private boolean isStarted;  
    private float speed;  
  
    public int getPower() {  
        return power;  
    }  
  
    public void start() {  
        – do something with hardware  
        isStarted = true;  
    }  
  
    public void speedUp(double v) {  
        if (isStarted) {  
            speed += v;  
        }  
    }  
}
```

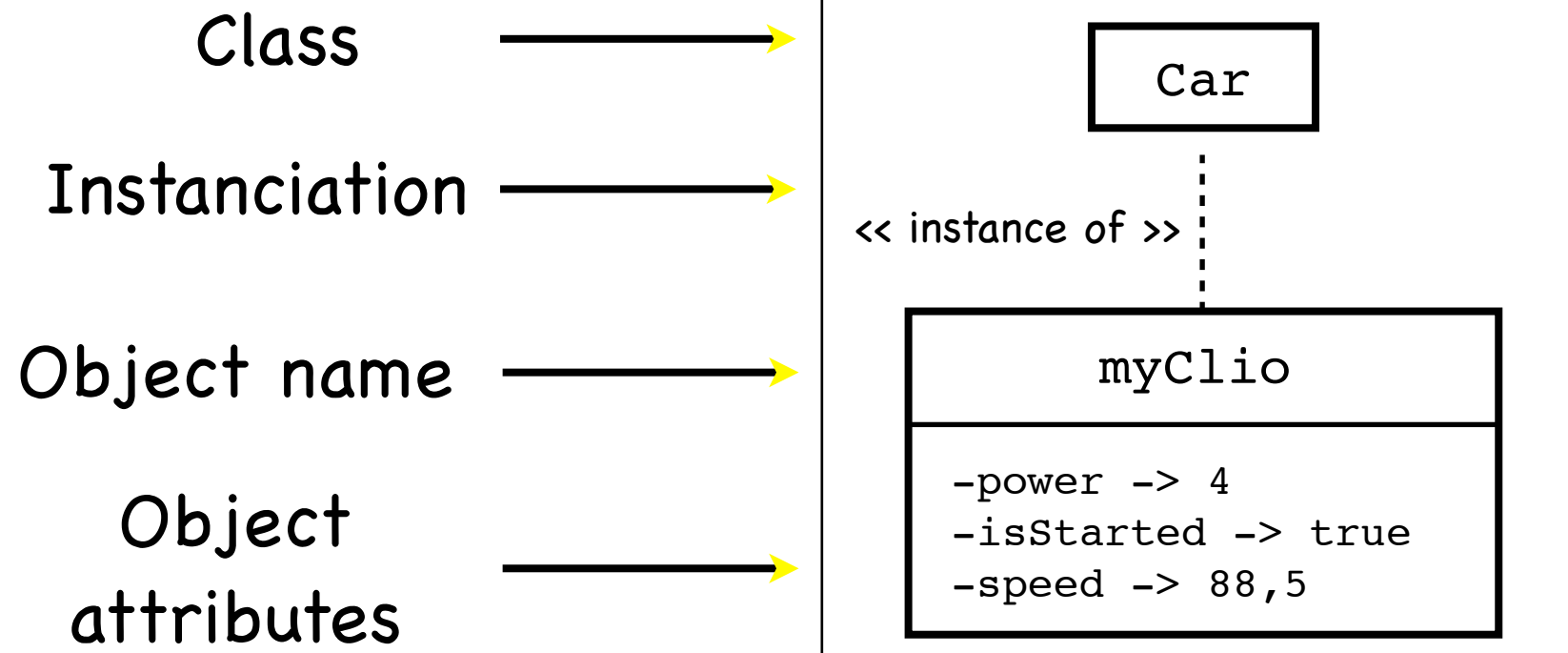


Reminder: the Object concept

- ☉ Objects are Class instances
- ☉ Objects are known through their reference
- ☉ Objects have a state





Reminder: the Object concept





Inheritance

Definition

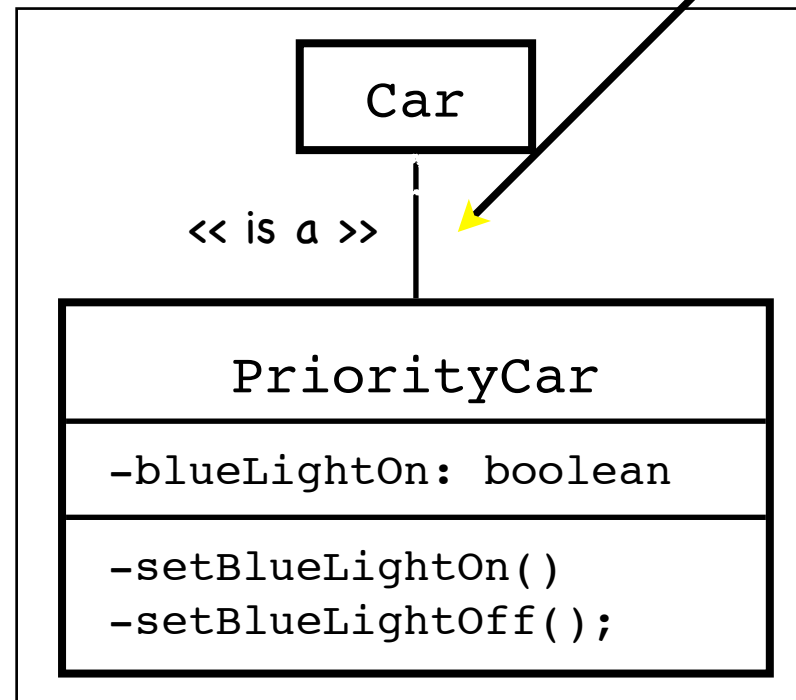
-  "In object-oriented programming (OOP), inheritance is a way to compartmentalize and reuse code by creating collections of attributes and behaviors called objects that can be based on previously created objects."
-  Wikipedia



Inheritance

UML Inheritance Relationship

- ☉ A priority car is a car...
- ☉ ... which owns a "blue light"

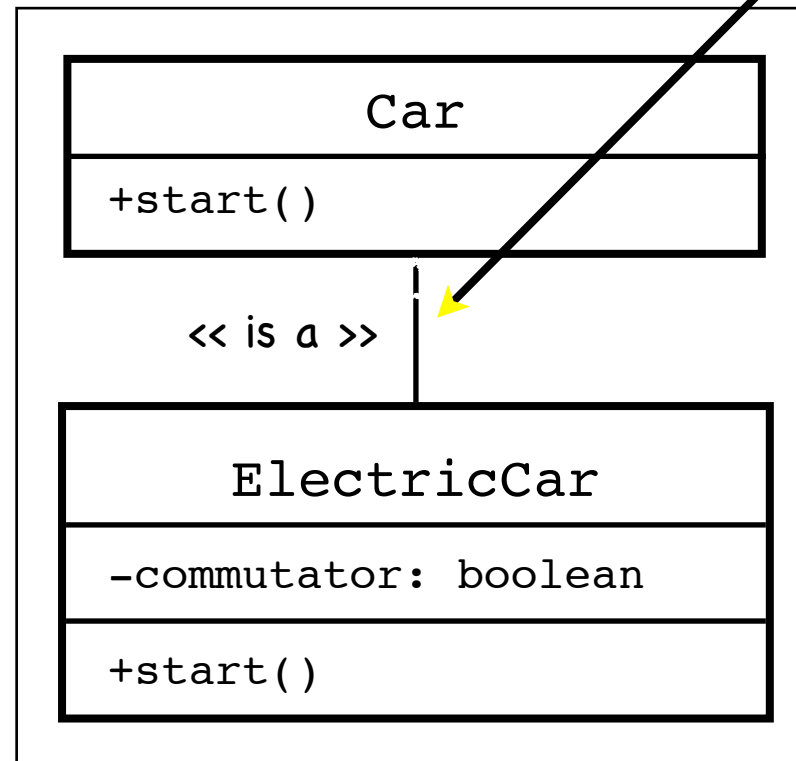




Inheritance

UML Inheritance Relationship

- An Electric Car is a car...
- ... which starts in a different way







Inheritance: Terminology

Inheritance

-  A PriorityCar inherits from Car

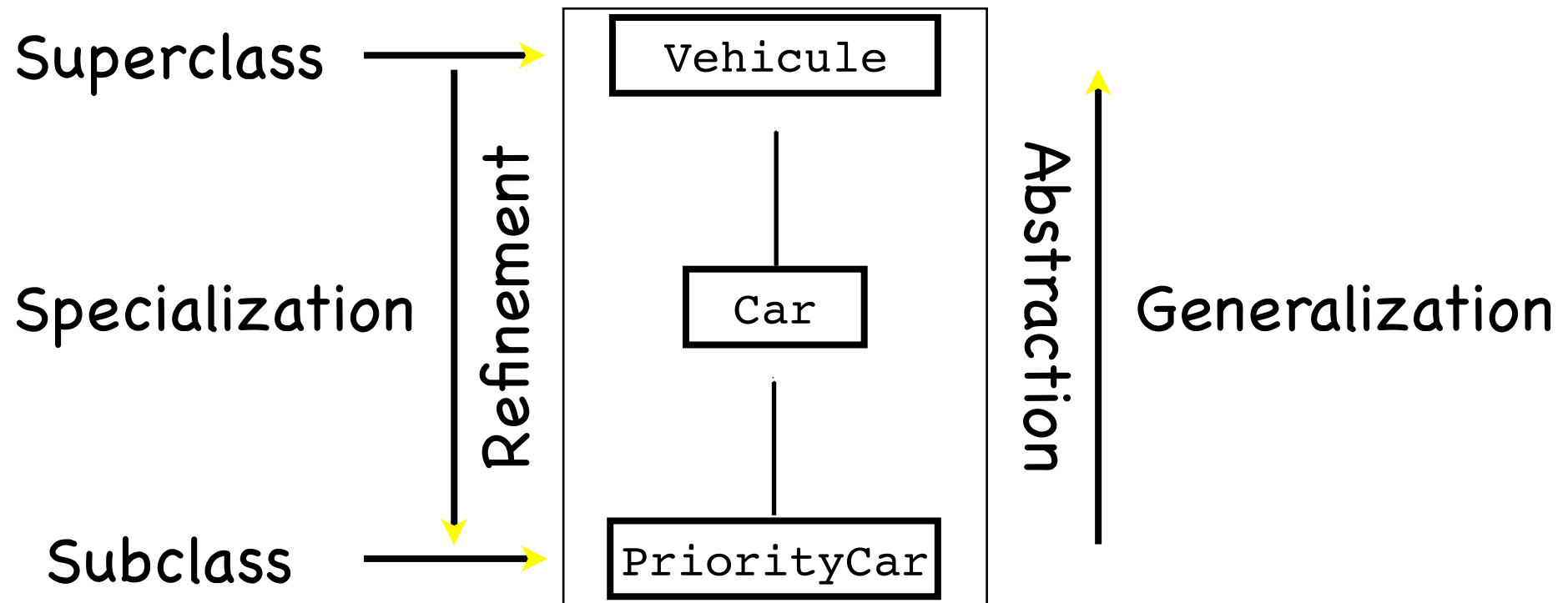
Parent classes, Child classes

-  Car is the Parent Class (also superclass) of PriorityCar
-  PriorityCar is a Child Class (also subclass) of Car



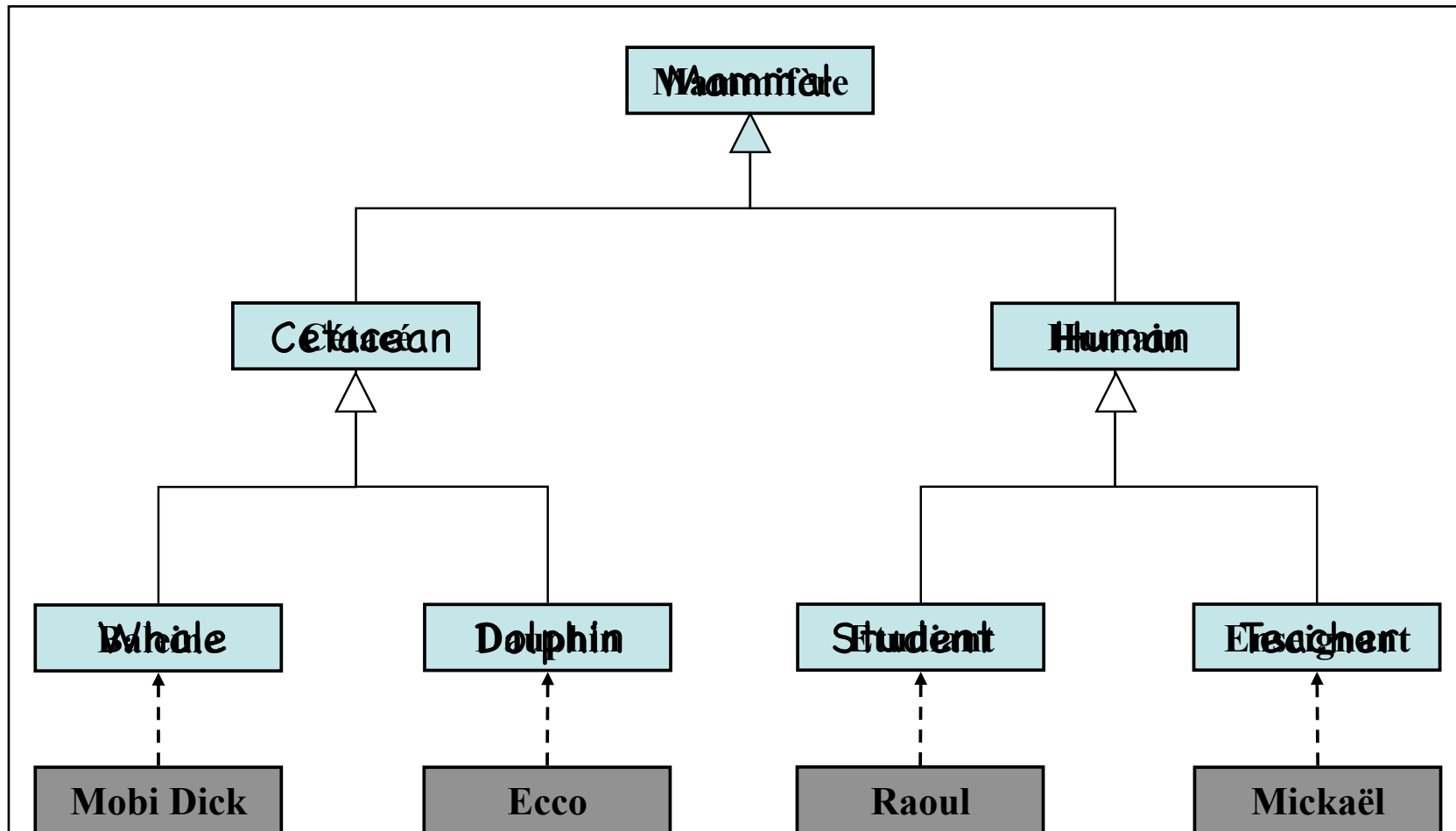
Inheritance: Specialization/Generalization

the "is a" relationship





Inheritance is "natural"

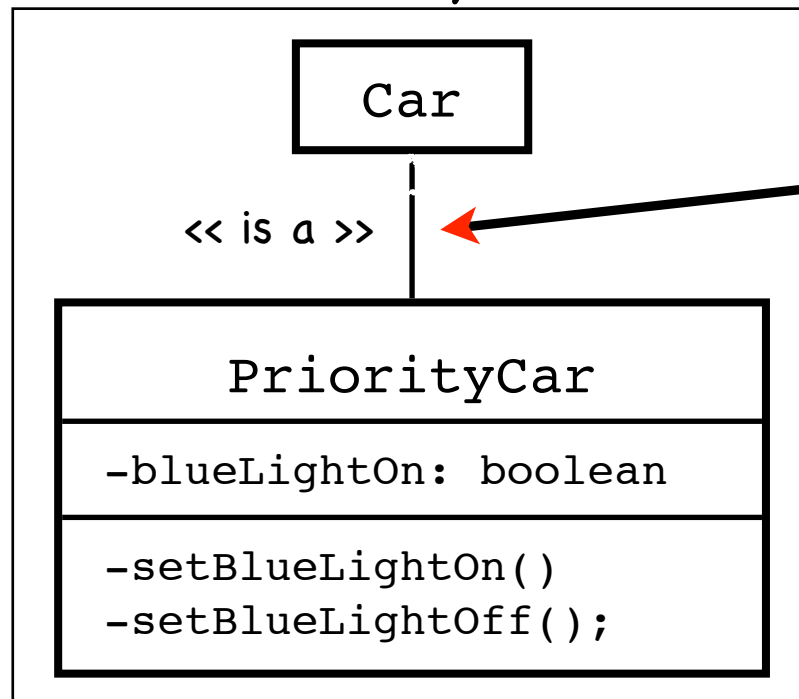




Simple inheritance

in Java

the keyword **extends**



```
public class PriorityCar
    extends Car {
    private boolean blueLightOn;

    public void setBlueLightOn() {
        - do something with hardware
        blueLightOn = true;
    }

    public void setBlueLightOff() {
        - do something with hardware
        blueLightOn = false;
    }
}
```