

Object Oriented Approach

Chapter 5

Dr. Youssef ATAT

Inheritance/Héritage

- Create new class from existing class
 - Absorb existing class's data and behaviors
 - Enhance with new capabilities
- Generalization allows:
 - To share the most general knowledge in **super-classes** (*base classes*)
 - To define increasingly specific information by defining specialized classes named *subclasses* (or *derived classes*)
- The inheritance is a concept attached to the classes which support generalization

Inheritance = build a class from other (others)

Inheritance

- How does it work?
 - A subclass inherits the attributes and the methods of its super-class
 - Specialization is done by:
 - Enhancement (new attributes, new operations)
 - Redefinition/Substitution (attributes, operations)
- The inheritance relation is:
 - Transitive
 - Realized with an aim of re-use
 - Can be simple or multiple

Inheritance

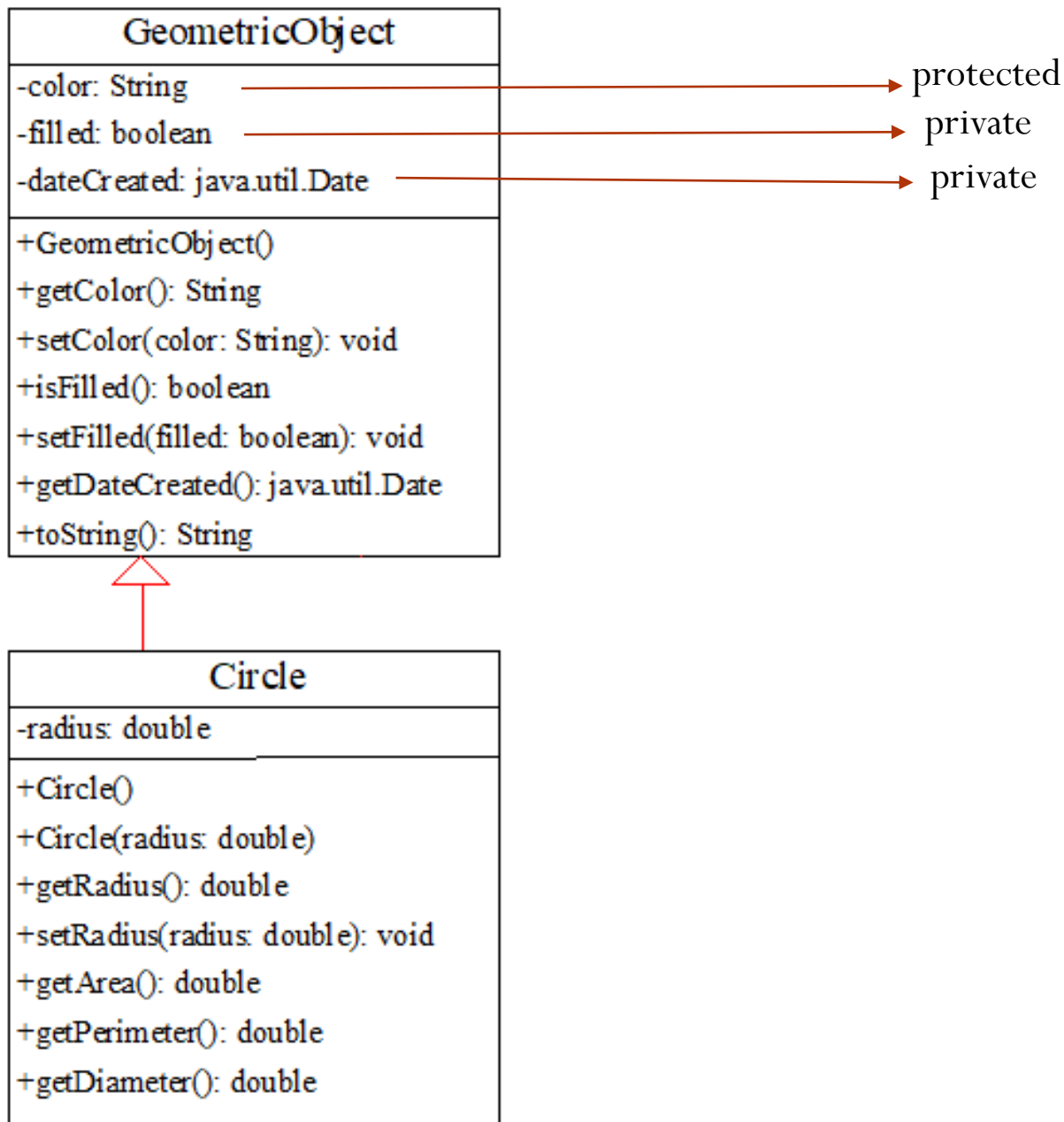
- “is-a” vs. “has-a”
 - “is-a”
 - Inheritance
 - subclass object treated as superclass object
 - Example: Car *is a* vehicle
 - Vehicle properties/behaviors also car properties/behaviors
 - “has-a”
 - Composition
 - Object contains one or more objects of other classes as members
 - Example: Car *has* wheels

Superclasses and Subclasses

- Superclasses and subclasses
 - Object of one class “is an” object of another class
 - Example: a car is a vehicle
 - superclass: Vehicle
 - Subclass: car
- Superclass typically represents larger set of objects than subclasses

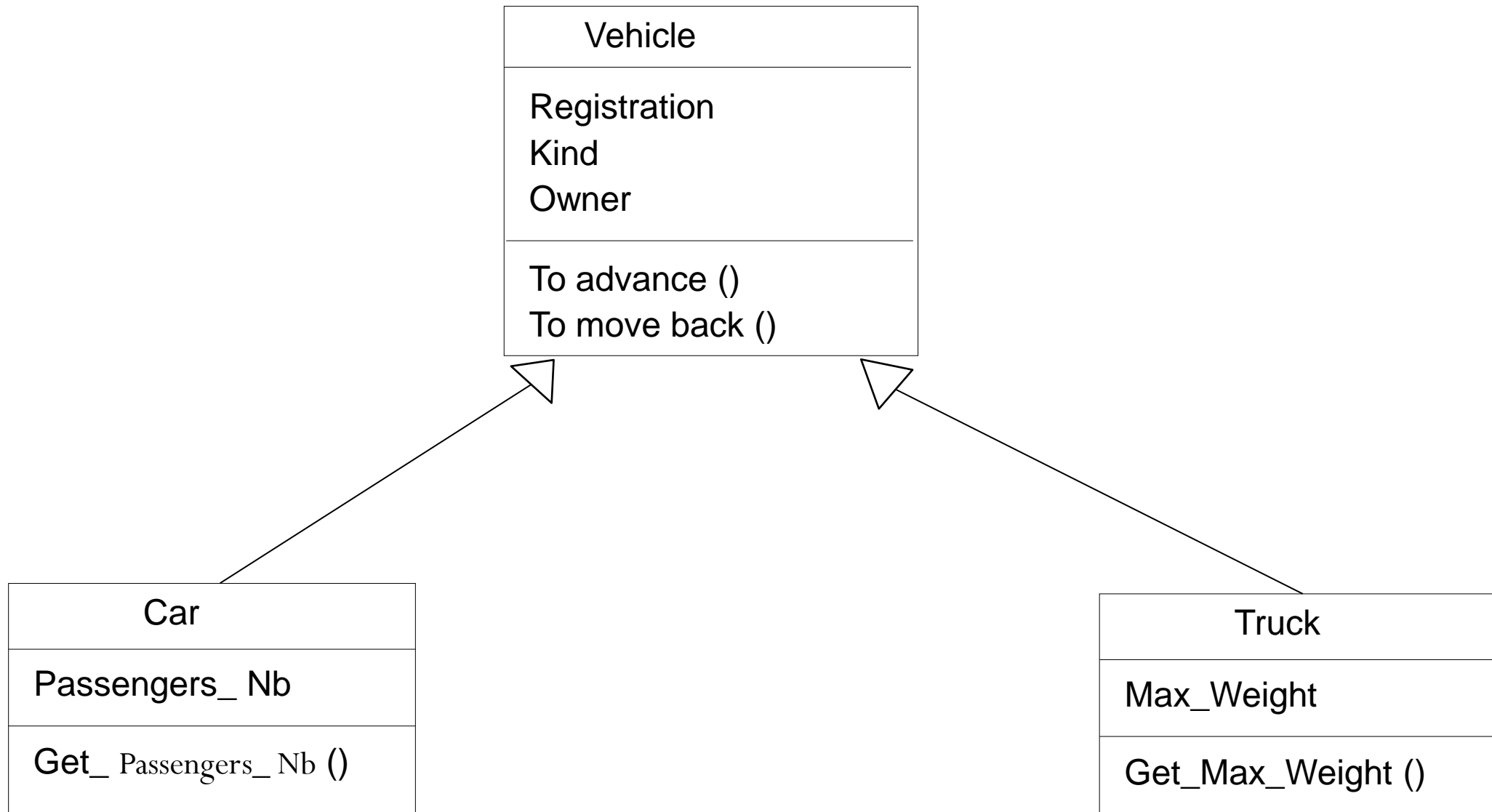
protected Members

- Intermediate level of protection between **public** and **private**
- **protected** members accessible to
 - superclass members
 - subclass members
 - Class members in the same package
- Subclass access superclass member
 - Keyword **super** and a dot (.)



Accessibility

Modifier on members in a class	Accessed from the same class	Accessed from the same package	Accessed from a subclass	Accessed from a different package
public	✓	✓	✓	✓
protected	✓	✓	✓	-
default	✓	✓	-	-
private	✓	-	-	-



Inheritance(Java)

- **extends**
- class Car **extends** Vehicle {}
- a superclass's constructors are not inherited in the subclass
- They can only be invoked from the subclasses' constructors, using the keyword super.
- *If the keyword super is not explicitly used, the superclass's no-arg constructor is automatically invoked*
- A constructor may invoke an overloaded constructor or its superclass's constructor. If none of them is invoked explicitly, the compiler puts super() as the first statement in the constructor.

Using the Keyword `super`

- The keyword `super` refers to the superclass of the class in which `super` appears. This keyword can be used in two ways:
 - To call a superclass constructor
 - To call a superclass method