

Introduction to Object-Oriented Programming

- Objects and classes
- Encapsulation and information hiding
- Mental exercises
 - Classification and exemplification
 - Aggregation and decomposition
 - Generalization and specialization

Objects and Classes



Mammal
Two-legs
Very large brains
Omnivorous (plants + meat)



Mammal
Four legs
Herbivorous (plant eater)

The Object Concept

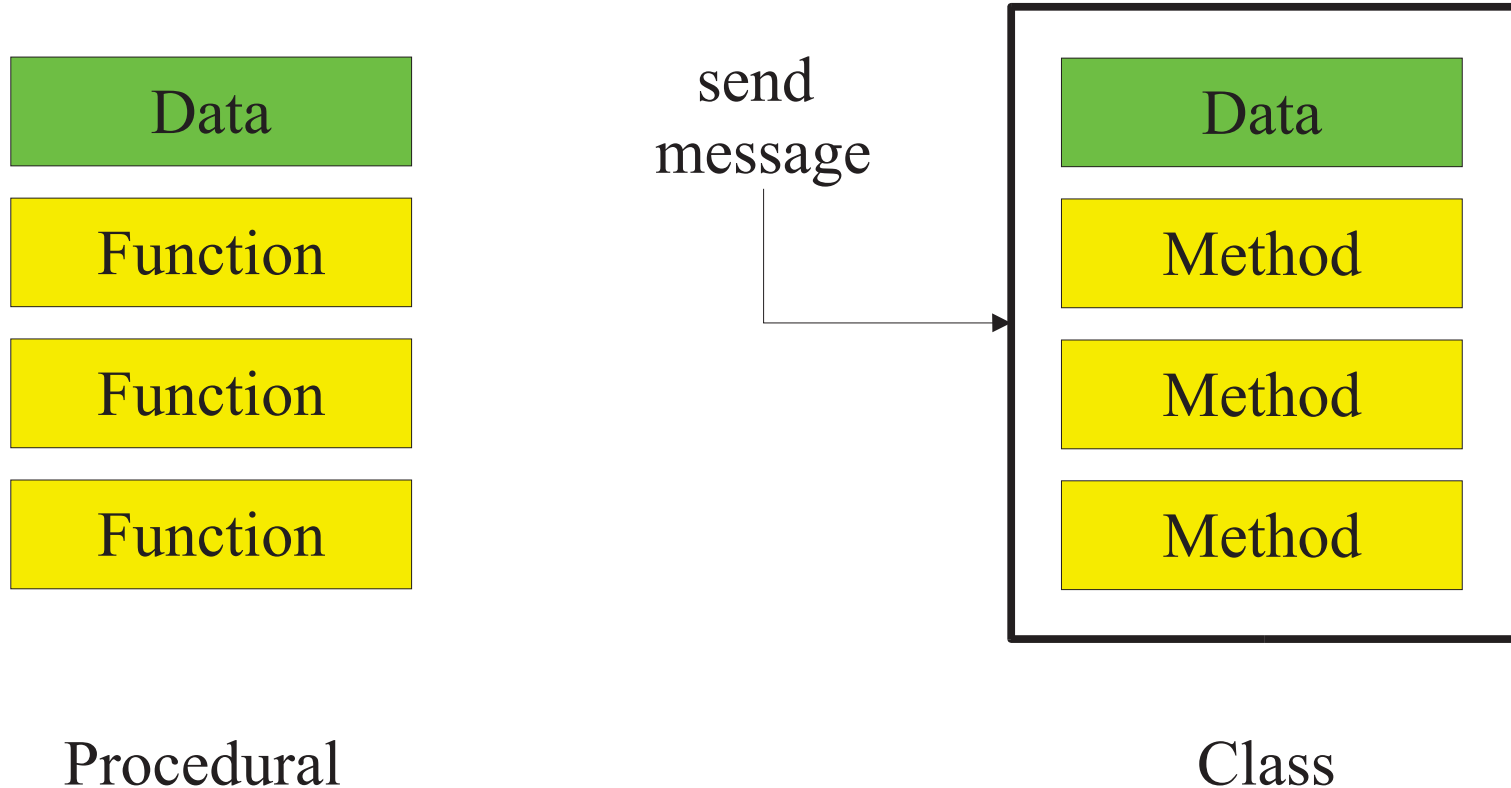
- An object is an *encapsulation* of data.
- An object has
 - identity (a unique reference)
 - ◆ social security number (cpr), employee number, passport number
 - state, also called characteristics (variables)
 - ◆ hungry, sad, drunk, running, alive
 - behavior (methods)
 - ◆ eat, drink, wave, smile
- An object is an instance of an *class*.
 - A class is often called an *Abstract Data Type (ADT)*.

The Class Concept

- A class is a collection of *objects* (or *values*) and a corresponding set of *methods*.
- A class encapsulates the data representation and makes data access possible at a higher level of abstraction.
- Example 1: A set of vehicles with operations for starting, stopping, driving, get km/liter, etc.
- Example 2: A time interval, start time, end time, duration, overlapping intervals, etc.

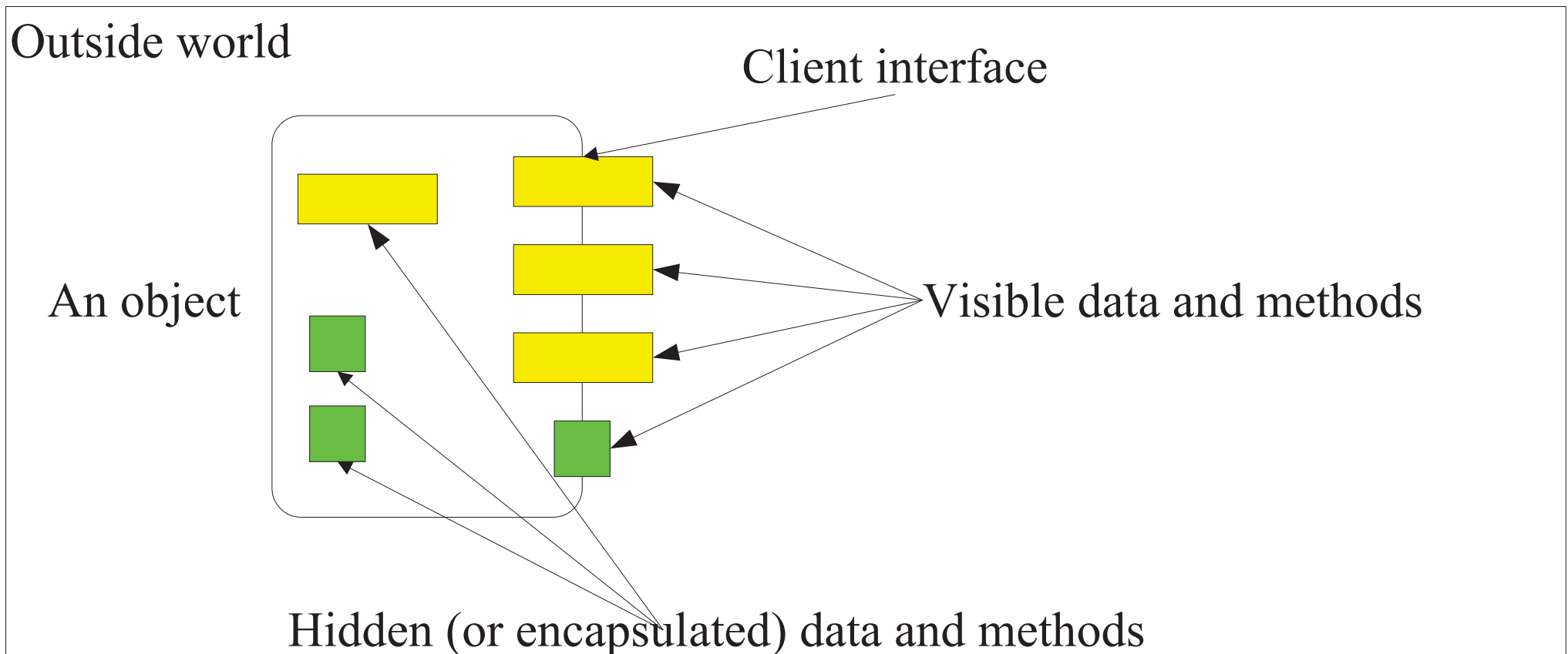
Encapsulation and Information Hiding

- Data can be encapsulated such that it is invisible to the “outside world”.
- Data can only be accessed via methods.



Encapsulation and Information Hiding, cont.

- What the “outside world” cannot see it cannot depend on!
- The object is a “fire-wall” between the object and the “outside world”.
- The hidden data and methods can be changed without affecting the “outside world”.



Class vs. Object

Class

- A description of the *common properties* of a set of objects.
- A concept.
- A class is a part of a program.

- Example 1: Person

- Example 2: Album

Object

- A representation of the *properties* of a single instance.
- A phenomenon.
- An object is part of data and a program execution.

- Example 1: Bill Clinton,

- Example 2: A Hard Day's

Connection between Object and Class

- In object-oriented programming we write classes
 - The text files we create contain classes!
 - Static
 - “One”
- Objects are created *from* classes
 - A class contains a “receipe” on how to make objects
 - Dynamic
 - “Many”

Ingrediens

250 g digestive biscuits food processor
125 g soft brown sugar saucepan
125 g butter wooden spoon
50 g raisins 18 cm sandwich tin (greased)
3 tablespoons cocoa powder fridge
1 egg, beaten knife
25 g = 1 oz
2.5 cm = 1 inch

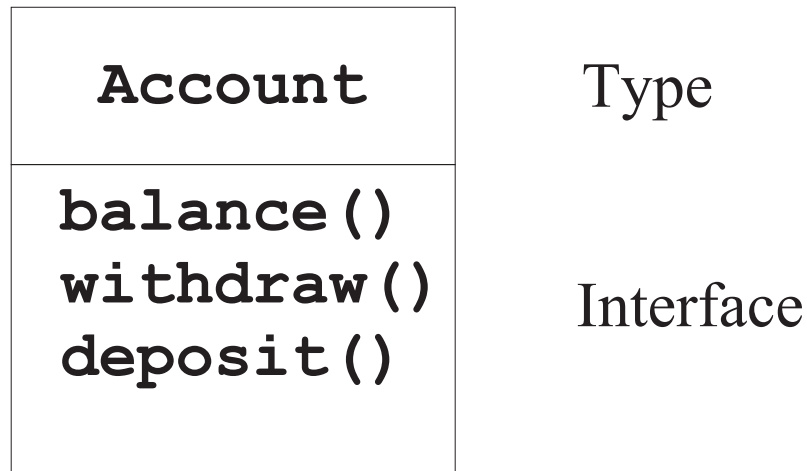
Process

blend
bake



Type and Interface

- An object has type and an interface.



- To get an object
Account a = new Account ()
Account b = new Account ()
- To send a message
a.withdraw ()
b.deposit ()
a.balance ()

Instantiating Classes

- An instantiation is a mechanism where objects are created from a class.
- Always involves storage allocation for the object.
- A mechanism where objects are given an initial state.

Static Instantiating

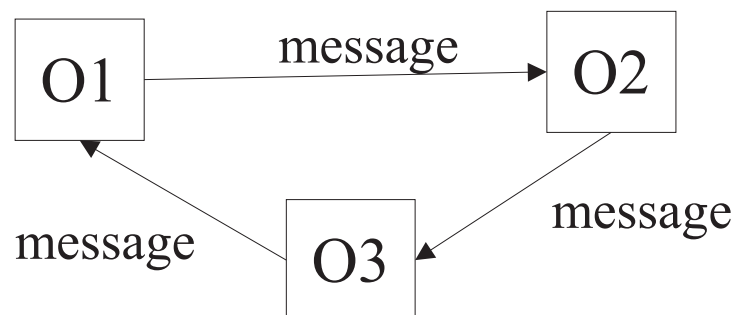
- In the declaration part of a program.
- A static instance is implicitly created

Dynamic Instantiating

- In the method part of a program.
- A dynamic instance is created explicitly with a special command.

Interaction between Objects

- Interaction between objects happens by *messages* being send.
- A message activates a method on the calling object.
- An object O1 interacts with another object O2 by calling a method on O2 (must be part of the client interface).
 - “O1 sends O2 a message”
- O1 and O2 must be *related* to communicate.
- The call of a method corresponds to a function (or procedure) call in a non-object-oriented language such as C or Pascal.



Phenomenon and Concept

- A *phenomenon* is a thing in the “real” world that has individual existence.
 - an object
- A *concept* is a generalization, derived from a set of phenomena and based on the common properties of these phenomena.
 - a class
- Characteristics of a concept
 - A name
 - *Intension*, the set of properties of the phenomenon
 - *Extension*, the set of phenomena covered by the concept.

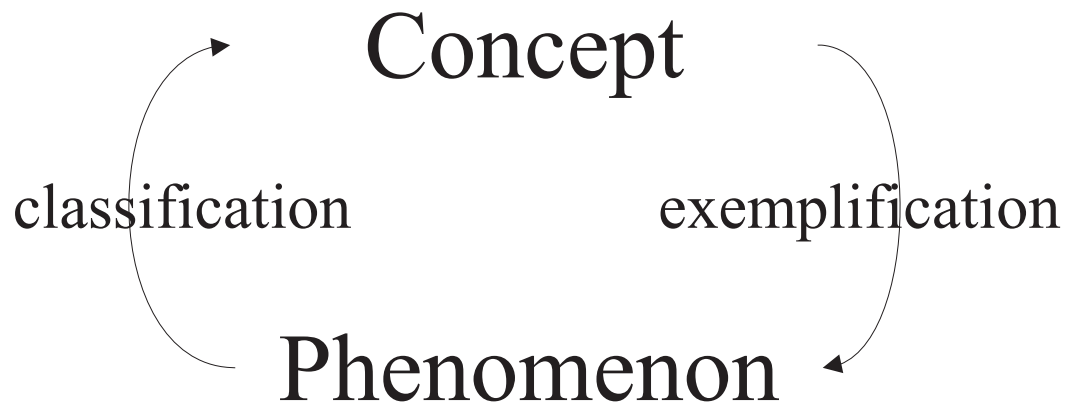
Classification and Exemplification, Examples

- hat, 23, 34, mouse, telephone, book, 98, 45.34, hello
 - numbers: 23, 34, 98, 45.34
 - words: hat, mouse, telephone, book, hello

- mouse, tyrannosaurus rex, allosaurus, elephant, velociraptor
 - dinosaur: tyrannosaurus rex, allosaurus, velociraptor
 - mammal: mouse, elephant

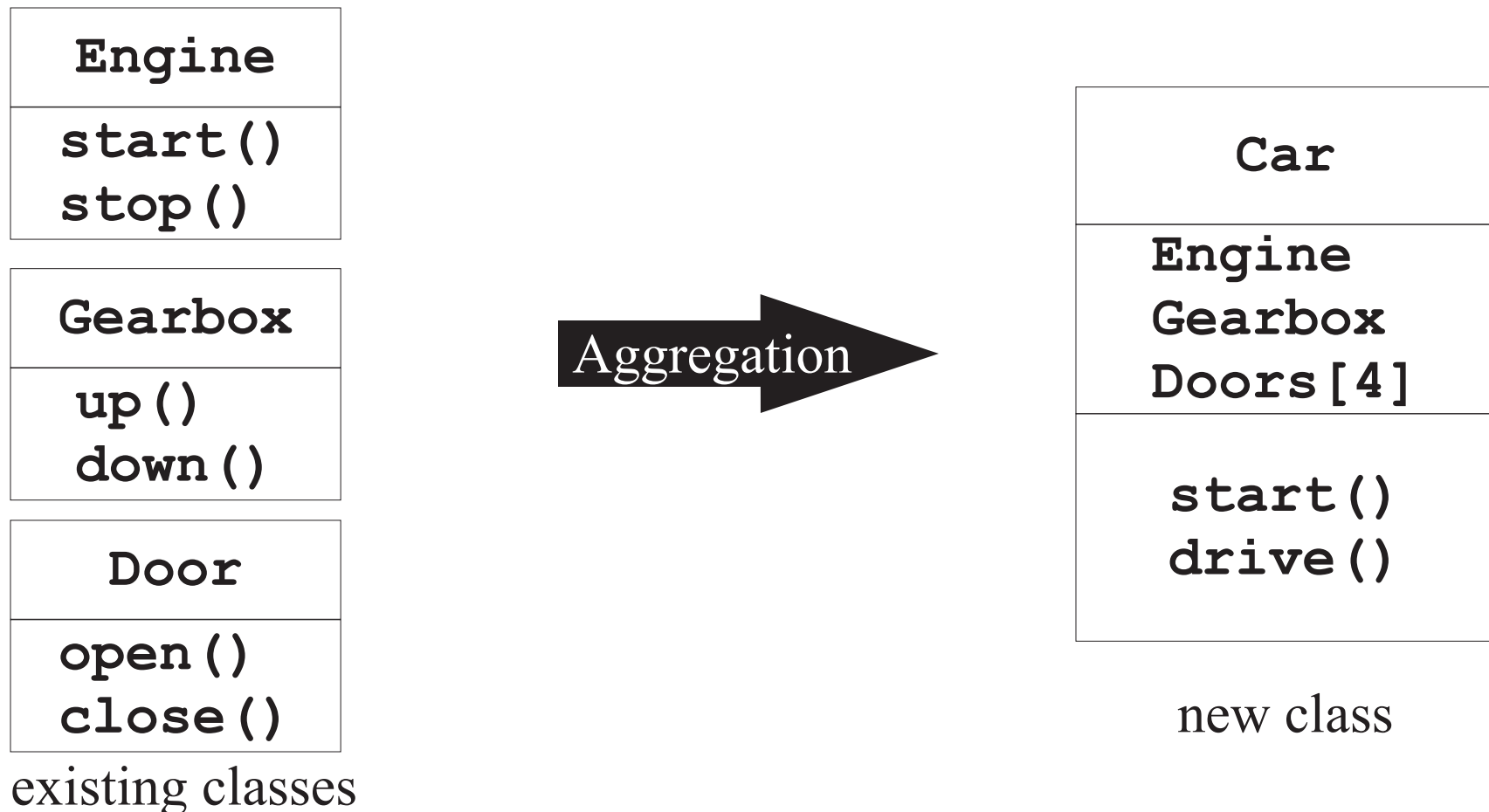
Classification and Exemplification, cont.

- A *classification* is a description of which phenomena that belongs to a concept.
- An *exemplification* is a phenomenon that covers the concept



Aggregation and Decomposition, Example

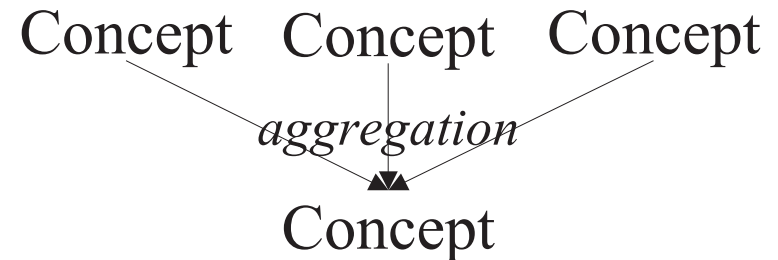
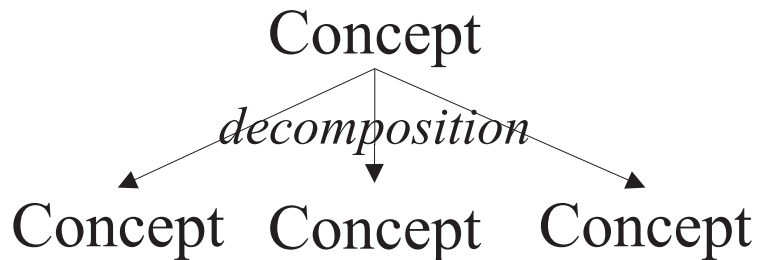
- Idea: make new objects by combining existing objects.
- *Reusing the implementation!*



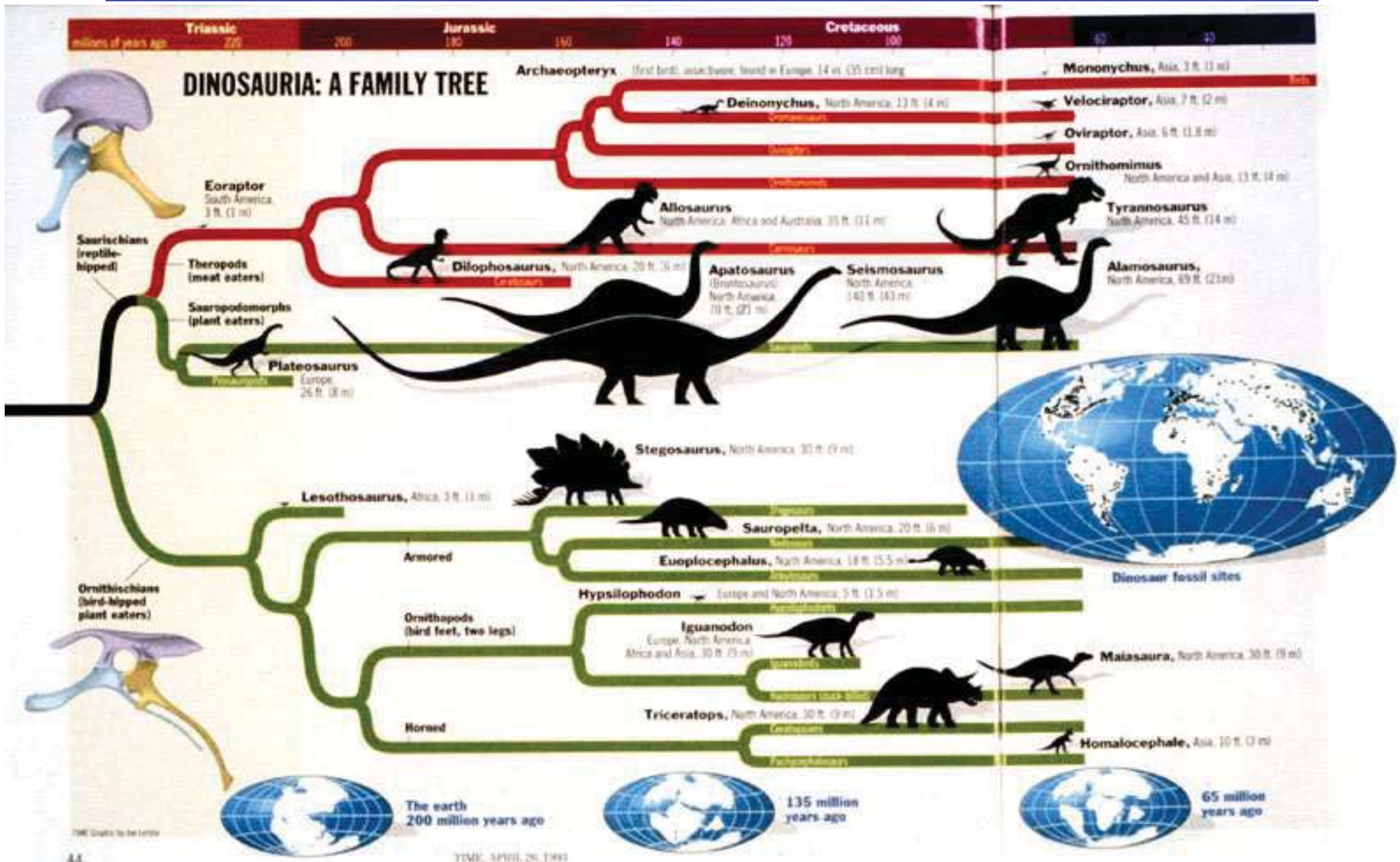
- **Car “has-a” Gearbox and Car “has-an” Engine**

Aggregation and Decomposition

- An *aggregation* consists of a number of (sub-)concepts which collectively is considered a new concept.
- A *decomposition* splits a single concept into a number of (sub-)concepts.



Generalization and Specialization



Generalization and Specialization, cont.

- *Generalization* creates a concept with a broader scope.
- *Specialization* creates a concept with a narrower scope.
- *Reusing the interface!*

