Object Oriented Approach

Dr. Youssef Atat

Syllabus

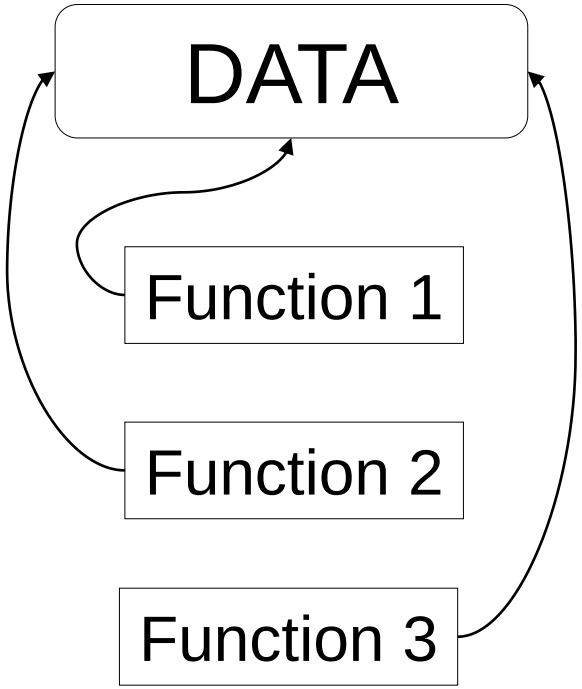
- Introduction about OO
- Principles of OO
- Object, Class
- Relationships between classes
- Java
- Abstract class
- Files
- Interface: Cloneable, Serializable

Introdcution

- Why OO?
  - Procedural programming
  - OO

Procedural programming

program = a succession of instructions carried out by a machine. These instructions act on data.

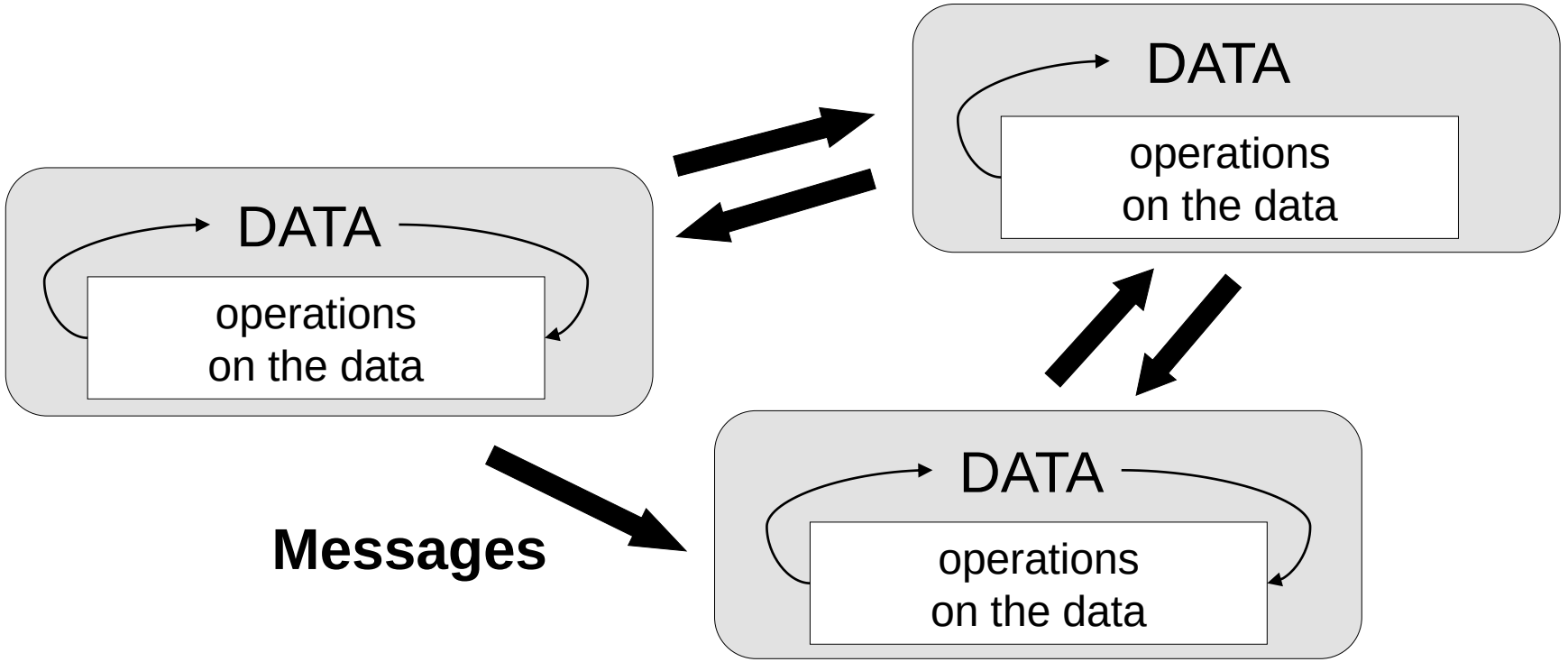


Procedural programming

- Functions and procedures act remotely on the data
- The Accent is laid on the actions: **What** do we want to make?
- Dissociation between functions and data: problem of Data Structures modification
- Functions call other functions and can modify the data.
- No data protection

Object Oriented

- A program = a society of entities. These entities collaborate to solve the final problem by sending messages.



Object Oriented

- An entity = an object which takes into account its own management (responsible object)

Object = data (attributes) + operations on these data (methods)

- Encapsulation: regrouping code and data

Concepts of Object Oriented

Object

Object = data (attributes) + operations on these data (methods)
= entity in the application domain of the problem

- Object Example: The blue Renault Megane matriculated 123 456 B

Object Renault_Megane

Type: Renault
Category: Megane
Owner: Patrick
Matriculation: 123 456 B
Color: Blue

Attributes

Start ()
Move backward ()
Move forward ()

Methods

Object

- Another example

red Renault CLIO registered 555 444 B

Object :Renault_Clio
Type: Renault
Category: Clio
Owner: Me
Matriculation: 555 444 B
Color: Red

Start ()
Move backward ()
Move forward ()

Class

Class= model describing the contents (attributes) and the behavior (methods) of the future objects of the class

= used to create objects

- Examples: the class of the vehicles, the class of the universities, the class of the students…
- A class: Description of a family/category **of *objects*** having
  - Same structure
  - Same behavior
- Class Automobile can be described by:

Class

Class automobile

Type

Category

Owner

Matriculation

Color

Start ()

Move backward ()

Move forward ()

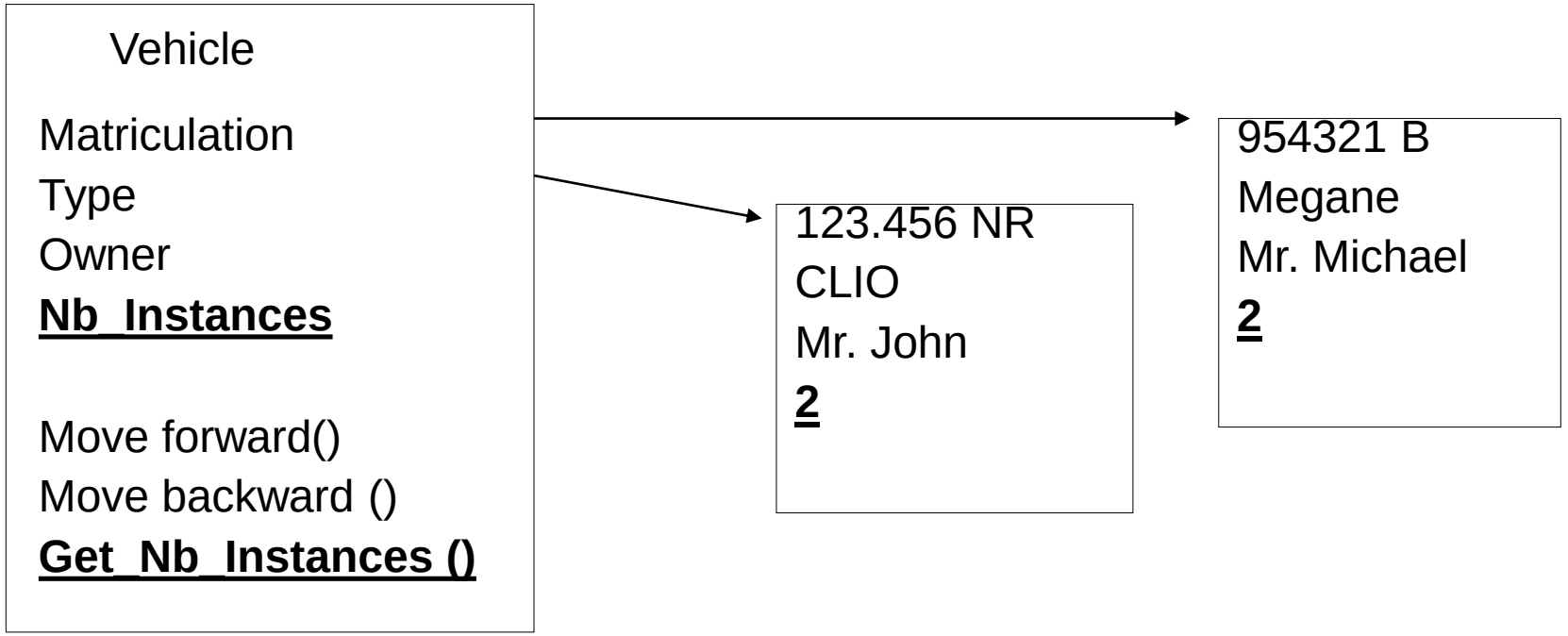
Instantiation

- The class is an abstraction of the common properties of a set of objects.

Object = instance of a class

Instantiation

- Variable of class: its value is shared by all the instances of the class
- Method. of class: related to class variables



Visibility

- public: the data or methods is visible to any class
- private: the data or methods can be accessed only by the declaring class
- protected: the data or methods is visible to the derived classes.

Encapsulation

- Each object is composed of two parts:
  - Operations on this object (public part)
  - Internal (close) part : sensible part of the object (private) to not be shown in public
- The users of this object (i.e external elements) should see only the public part
- In the OO world, the encapsulation has two directions:

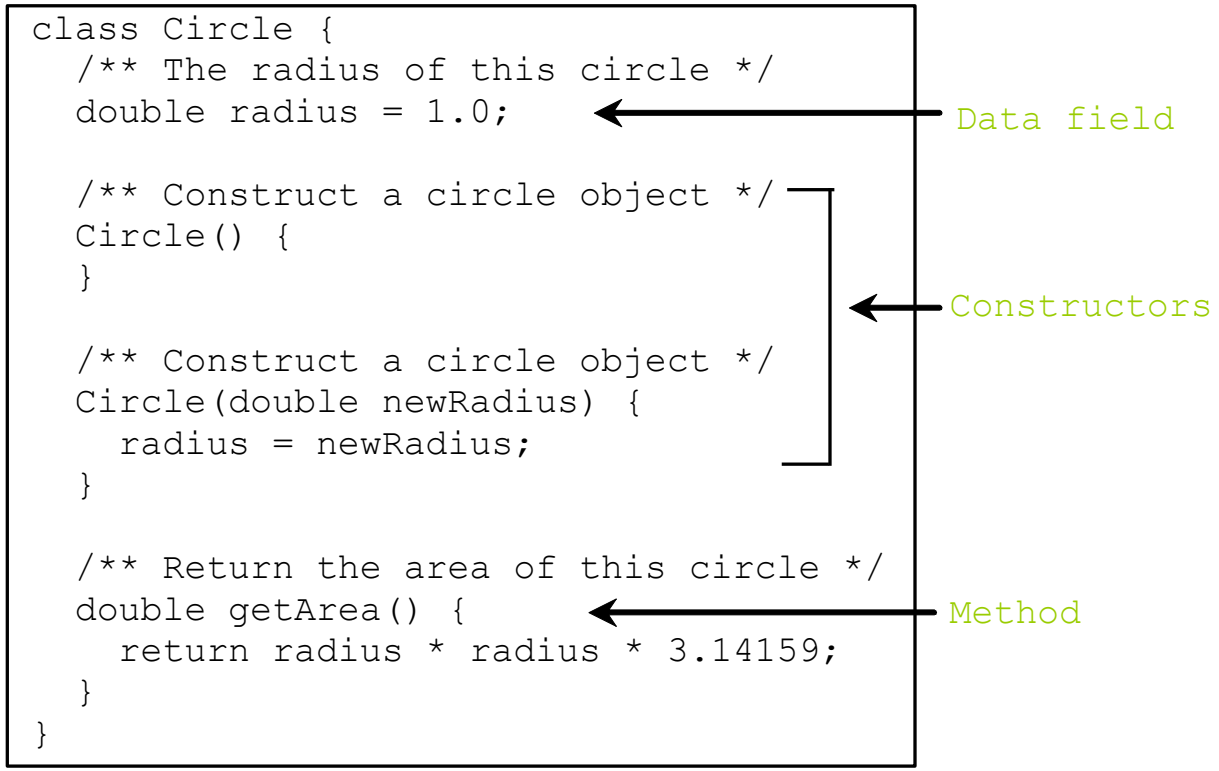
Encapsulation

Encapsulation =

1) Regrouping code and data
2) Masking information (data hiding)

Constructors

- a class provides a special type of methods, known as constructors, which are invoked
- to construct objects from the class.



Creating Objects Using Constructors

- `new Circle(); default constructor`
- `new Circle(5.0);`
- Example:
```
Circle c1 = new Circle();
Circle C2 = new Circle(5.0);
c1.getArea();
```
- In Java, the default constructor, is provided automatically *only if no constructors are explicitly declared in the class*.