032CM - 2025

PROGRAMMING FOR COMPUTATIONAL CHEMISTRY

Python classes

Gianluca Levi

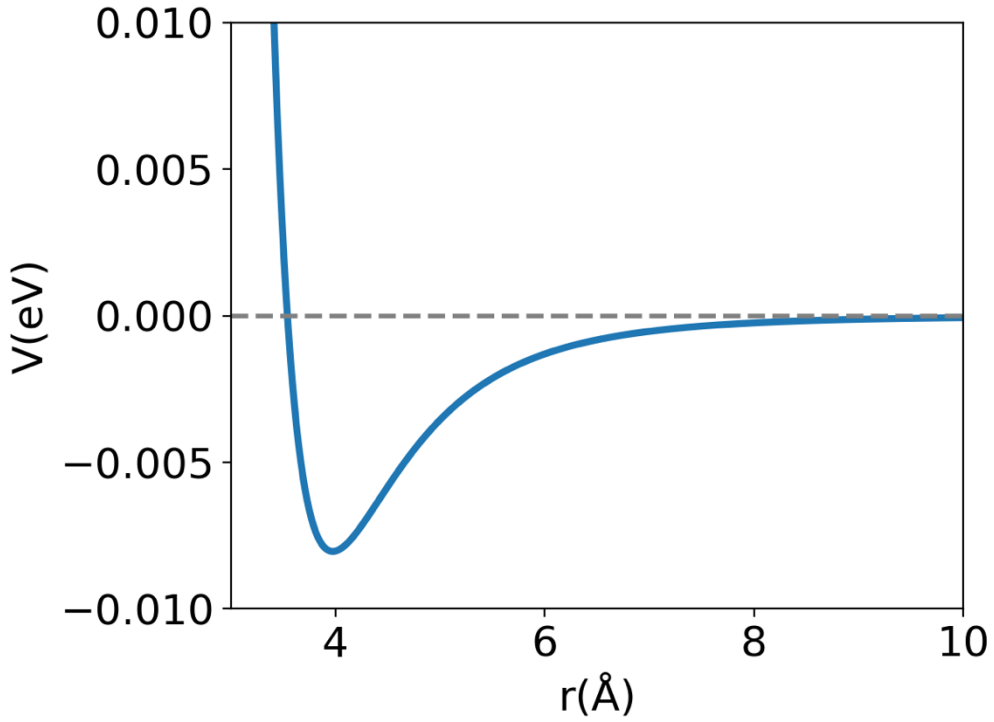
gianluca.levi@units.it, giale@hi.is

Office: Building C11, 3rd floor, Room 329

Fall 2025

Lennard-Jones potential

$$V(r) = 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right]$$



Simple **pairwise model** for interatomic interactions

Short-range repulsion + **dispersion attraction**

Used to model short-range interactions in **molecular dynamics simulations**

Specific vs. general implementation

A class is a **data type**

- Creates **new data type**
- Objects made from the class are **instances** of that type

Bundles data and behavior

- Packs data (attributes) and functions (methods) operating on them
- Avoids setting global variables (and related side effects)
- Optimal for multiple parameter sets
- Improves **reusability** and **readability**

General structure of a class

```
def Class_Name:
    """
    Description of the class
    """
    def __init__(self, p1, p2, ...):
        self.var1 = p1
        self.var2 = p2
            ...
            ...

    def function_name(self, fp1, ...):
        <expression>

        return result_1, result_2, ...
```

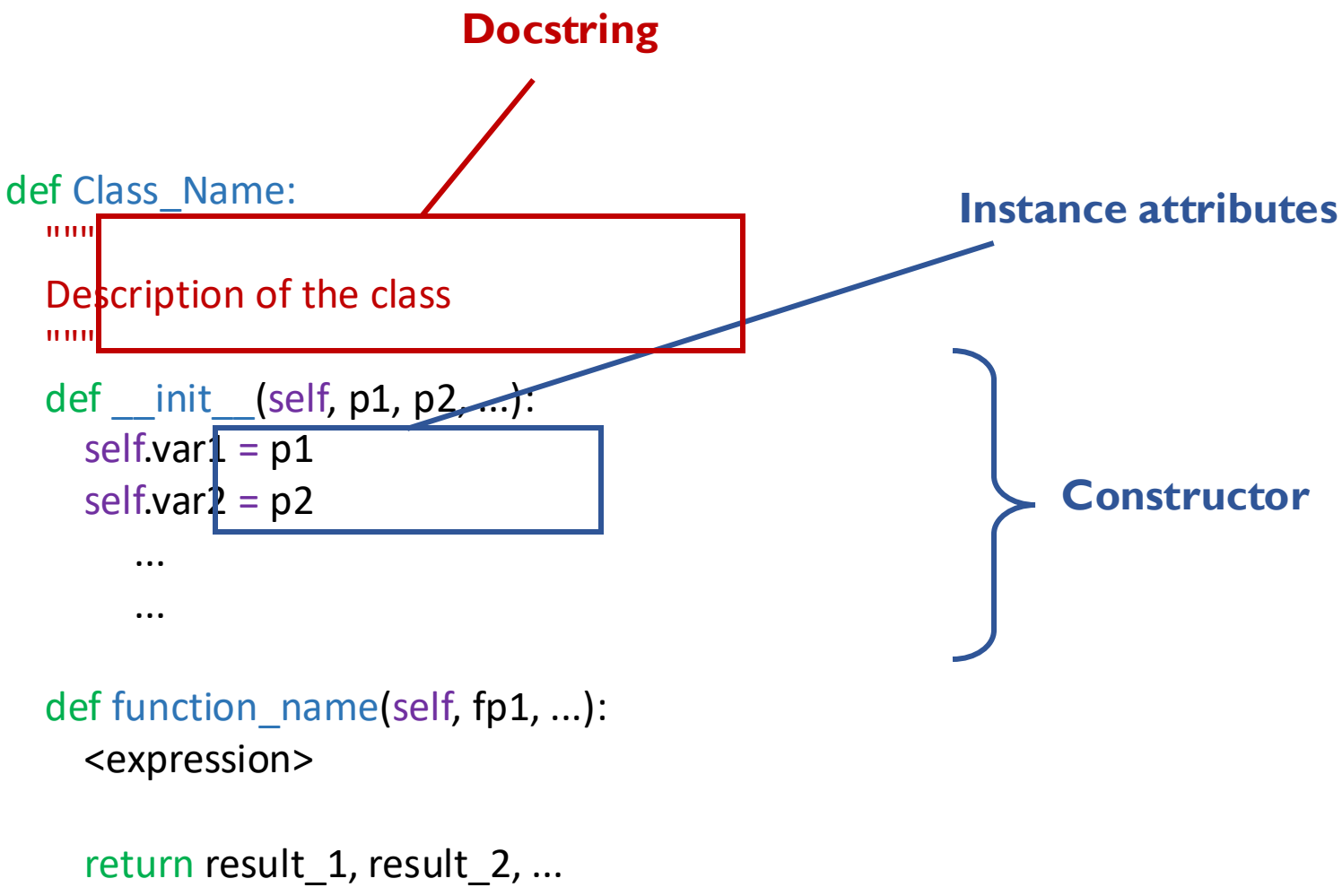
General structure of a class

Docstring

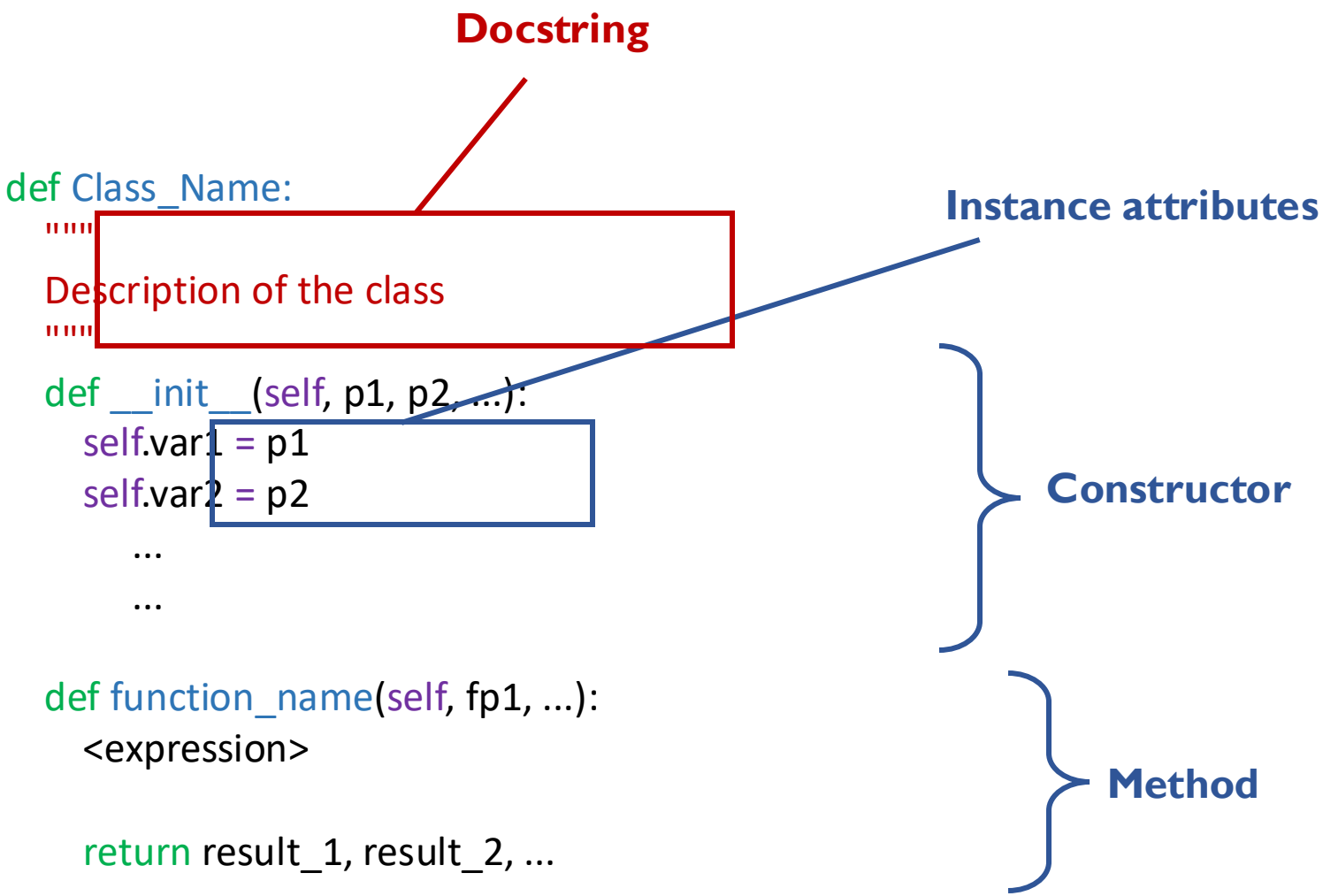
```
def Class_Name:
    """
    Description of the class
    """
    def __init__(self, p1, p2, ...):
        self.var1 = p1
        self.var2 = p2
            ...
            ...

    def function_name(self, fp1, ...):
        <expression>

        return result_1, result_2, ...
```

Constructor

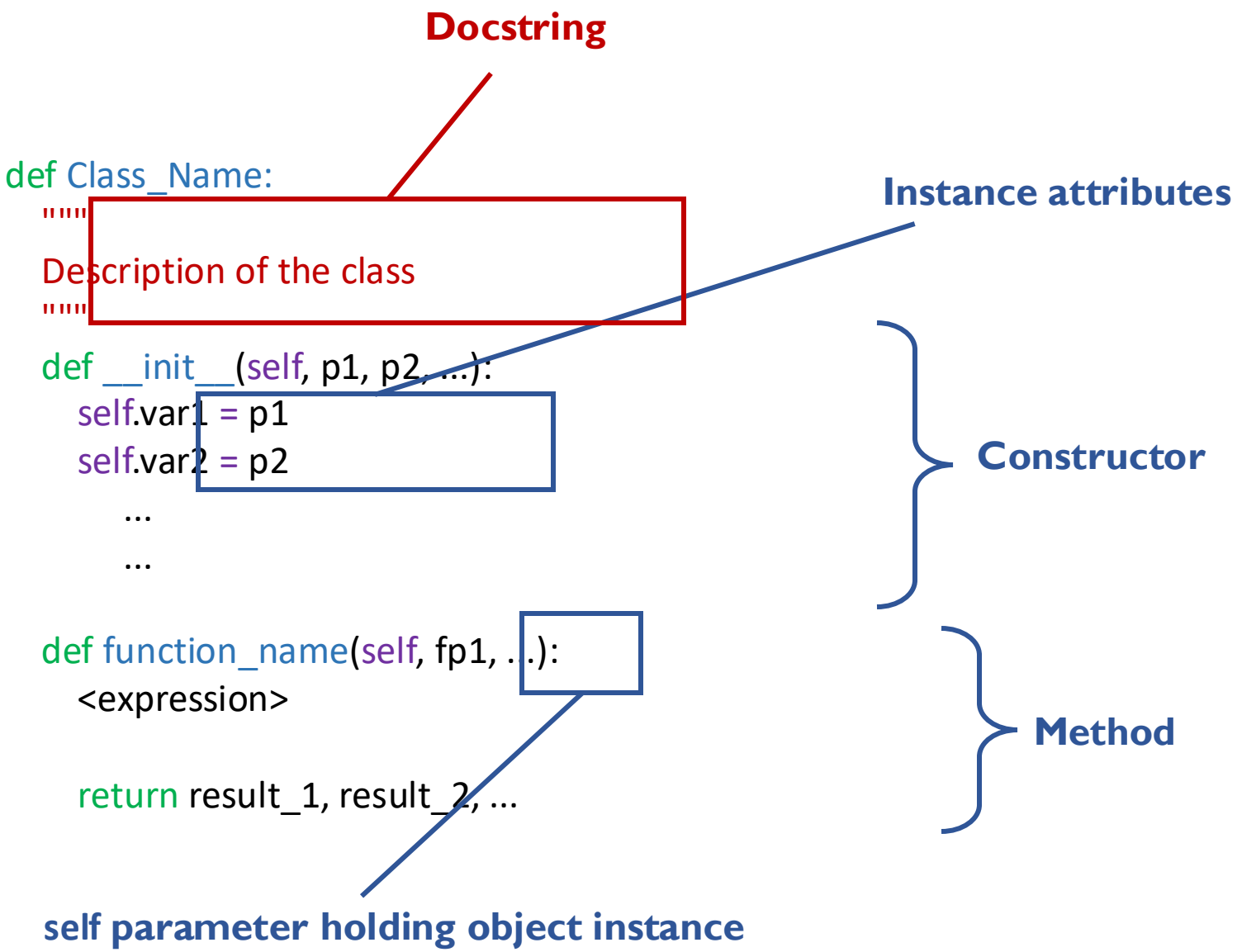
General structure of a class



General structure of a class



General structure of a class



Assignment 5

Problem 2

The paiwise Lennard-Jones potential can be used to model the interatomic interactions of noble-gas atoms. The Lennard-Jones interaction energy between atoms is given by:

$$E_{LJ} = \sum_{i=1}^{N} \sum_{j>i}^{N} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right]$$

where N is the number of atoms and r_{ij} the distance between atoms i and j. The positive r^{-12} term describes the repulsive part of the potential, while the negative r^{-6} term describes the attractive part.

(a) Create a Python class called `Atoms` that represents a cluster of atoms with attribute the (x, y, z) coordinates of the atoms in the cluster (NumPy array). Include a `set_positions` method to update the atom positions and a `get_pairwise_distances` method that computes and returns a vector of pairwise distances between all atoms.

(b) Create a `LennardJonesCalculator` class that computes the Lennard-Jones energy of the cluster of atoms with attributes `epsilon` and `sigma`. The class should have a `calculate_energy` method that takes the `Atoms` object and returns the total Lennard-Jones energy.

(c) Calculate the energy of the following two configurations (atomic coordinates in Å) of a cluster of four Ar atoms ($\epsilon = 8.0$ meV and $\sigma = 3.54$ Å).
Linear chain:

$$\texttt{positions} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 4.0 & 0.0 & 0.0 \\ 8.0 & 0.0 & 0.0 \\ 12.0 & 0.0 & 0.0 \end{bmatrix}$$

Square arrangement:

$$\texttt{positions} = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 4.0 & 0.0 & 0.0 \\ 0.0 & 4.0 & 0.0 \\ 4.0 & 4.0 & 0.0 \end{bmatrix}$$

Which configuration has the lowest energy?