

Review Object Oriented Programming

Exercise 1 : Polynomial

Create a class **Poly** which creates a polynomial.

- 1) Write the constructor of the class which will allow to define the attribute **coeff** , which will contain the list of coefficients of the polynomial in the order of increasing powers.
- 2) Write a method **degre()** which returns the degree of the polynomial.
- 3) Write a **derive()** method that returns the derived polynomial.
- 4) Write a method **value(x)** x being a number, which returns the evaluation of the polynomial when its variable is x.
- 5) Write a method **evalvalue(x)**, x being a number or a list of numbers, which returns the value(s) of the polynomial for each value of x.
- 6) Write a **display()** method to display the polynomial according to the following example.

```
In [11]: p=Poly([0,2,0,1,3])  
  
In [12]: p.affiche()  
2x+x^3+3x^4
```

- 7) Write a method to override the **__add__** method
- 8) Write a method to override the method **__str__**.
- 9) Write a method to overload the method **__mul__**.
- 10) Make the multiplication of a polynomial by a number work in both directions ($a \cdot P = P \cdot a$)
- 11) Write a method to overload the **__len__** method
- 12) Write a test for each method.

Exercise 2 : Genealogy

- 1) Create a class **Personne**
- 2) Write the constructor of the class which will take as parameters a name and a first name and store them as attributes of the class
- 3) Modify the constructor to initialize an attribute of class father to None and an attribute of class mother to None
- 4) Create a sender (a **set** method) to modify the father attribute
- 5) Create a sender (a **set** method) to modify the attribute mother
- 6) Create a **display** method to display the first and last name of the person and his parents. A bonus will be granted if the special cases are managed and if the parents are themselves objects of type Person.
- 7) Create a **Family** class with a constructor that takes a list of Person objects as parameter and stores it as class attribute
- 8) Create a method **add_member** in the Family class which takes as parameter a Person object and adds it to the list of the Family class

- 9) Create a method ***modify_member*** which takes as parameters the first and last name of the person to be modified and his parents and which modifies the parents of this person in the list of the class Family for this person
- 10) Create a person Jean ROCHE, a person Jacques ROCHE, a person Etienne LUCIEN, a person Sarah ROCHE and a person Béatrice COOL
- 11) Create the family ROCHE
- 12) Add a person Lucie ROCHE to the family ROCHE
- 13) Modify the parents of Lucie ROCHE