**Exercise 1: Properties, Descriptors and Operator Overloading.**

Let us consider a class Person with the following attributes: name, lastname, birthday with the obvious meaning and the corresponding setter and getters and the \_\_repr\_\_ to print it.

1. Extend the class Person in the class Student by adding a dictionary lectures with the lecture name as a key and the mark as a value, and the property grade\_average to calculate the marks average
2. Extend the class Person in the class Worker by adding an attribute pay\_per\_hour and the properties day\_salary, week\_salary, month\_salary, and year\_salary considering 8 working hours a day, 5 working days a week, 4 weeks a month, 12 months a year; note that to set one of the properties implies to recalculate the pay\_per\_hour value
3. Extend the class Person in the class Wizard by adding a property age that when used as a getter calculates the correct age in term of passed days from the birthday to the current day and when used as a setter it will change the birthday accordingly rejuvenating or getting old magically.

Repeat the exercise by using the descriptors instead of the properties.

**Exercise 2: Decorators.**

Let us consider a class MyMath with the following methods: fib, fact and taylor implementing the Fibonacci's series, the factorial and the Taylor's series for a generic function and a level of approximation respectively. Then implement the following decorators:

1. @memoization applied to a method stores in the class previously calculated results and reuses them instead of recalculating
2. @logging applied to a method writes on a file the method name, its actual arguments when a method is called (also by recursion)
3. @stack\_trace applied to a method prints its stack trace, i.e., the list of calls made to carry out the invocation.

**Exercise 3: Metaclasses.**

Let us consider the class Person again and implement the following metaclasses:

1. The metaclass Counter which counts how many times Person has been instantiated
2. The metaclass Spell that transforms the instances of Person in a Worker with the properties/descriptors of the previous exercise as real methods/attributes
3. The metaclass MultiTriggeredMethod that let activate a method only when called twice