

TUTORED & PRACTICAL WORK N°02

CLASSES AND OBJECTS

Exercise 1

Consider the following classes A, B, and C:

***** Class A *****

```
class A:
    def __init__(self):
        self.w = 1
        self._x = 2
        self.y = 3
        self.__z = 4
```

***** Class B *****

```
class B(A):
    def __init__(self):
        # print the values of w, x, y, z
```

***** Class C *****

```
class C:
    def __init__(self):
        #print the values of w, x, y, z
```

Complete the content of these classes in order of each class can access to the attributes of class A. Use the getters and setters to access the protected and private properties.

Exercise 2

1. Define a class **Book** with the following private attributes: title, author (Complete name), price.
2. Define the getters and setters of the private properties.
3. Define a constructor to initialize the attributes of the class with values entered by the user.
4. Define the method `display()` to display the information of the current book.
5. Write a program `TW_2_2.py` to create an object which is an instance of the class **Book**, make a balance of 50% of its price, then display the book information.

Exercise 3

1. Define a class `Client` with the following private attributes: `CIN`, `firstName`, `lastName`, `tel`.
2. Define a constructor to initialize all attributes with default empty value of `tel`.
3. Define the access methods to the different attributes of the class (getters & setters).
4. Define the `display()` method to display the `Client` information.
5. Create a class `Account` characterized by private attributes, its balance and a code that is incremented at its creation and its owner who represents a customer of the class `Client`.
6. Define a constructor allowing to create an account by indicating its owner and initializing the balance to 0 and the code in a sequential way.
7. Define the access methods to the different attributes of the class.
8. Add the following methods to the `Account` class:
 - A method to `credit()` the account, taking an amount in parameter.
 - A method to `credit()` the account, taking an amount and an account as parameters, crediting the current account and debiting the account passed as parameter.
 - A method to `debit()` the account, taking an amount in parameter.
 - A method to `debit()` the account, taking an amount and a bank account as parameters, debiting the current account and crediting the account passed as parameter.
 - A method to display the summary of an account.
 - A method to display the number of accounts created.
9. Create a test program `TW_2_3.py` for the `Account` class.