

# Object-Oriented Programming

Lab 3  
ENSIA 2021/2022

## Exercise 1 (15 minutes)

Given the class `Point` defined below, write an independent function `display`, which is a friend function of the class `Point` that displays the coordinates of a point. Write a program to create a point and display its coordinates.

```
1 class Point {  
2     public:  
3         Point (int abs=0, int ord=0) {  
4             x=abs; y=ord;  
5         }  
6  
7     private:  
8         int x,y;  
9 };
```

## Exercise 2 (30 minutes)

In Exercise 2, a point has been represented by integer coordinates. In a way a little different, we can also represent points with rational coordinates. So a point may be considered as the composition of two `Rational` (class `Rational` of Lab 1 Exercise 4).

A segment can be described by its extremities (objects of the class `Point`).

1. Create a class called `Point`. Use `Rational` objects to represent the `private` data for the class. Provide a constructor function that enables an object of this class to be initialised when it is declared. The constructor should contain default values in case no initialisers are provided. Provide `public` member functions to get and set the coordinates of a point.
2. Create a class called `Segment`. Use `Point` objects to represent the `private` data for the class. Provide a constructor function, member functions to get and set the extremities of a segment and `public` function to get the dimension of a segment.

Write a driver program to test your `Point` and `Segment` classes.

## Exercise 3 (35 minutes)

Create class `IntegerSet` for which each object can hold integers in the range 0 through 100. A set is represented internally as an array of ones and zeros. Array element `a[i]` is 1 if integer

`i` is in the set. Array element `a[j]` is 0 if integer `j` is not in the set. The default constructor initializes a set to the so-called “*empty set*”, *i.e.* a set whose array representation contains all zeros.

Provide member functions for the common set operations. For example:

- provide `unionOfSets` member function that creates a third set that is the set-theoretic union of two existing sets (*i.e.* an element of the third set array is set to 1 if that element is 1 in either or both of existing sets, and an element of the third set array is set to 0 if that element is 0 in each of the existing sets).
- provide `intersectionOfSets` member function that creates a third set that is the set-theoretic intersection of two existing sets.
- provide `insertElement` member function that inserts a new integer `k` into a set (by setting `a[k]` to 1).
- provide `deleteElement` member function that deletes a new integer `m` (by setting `a[m]` to 0).
- provide `inputSet` member function that puts elements given by the user into the set.
- provide `printSet` member function that prints a set as a list of numbers separated by spaces. Print only those elements that are present in the set. Print --- for an empty set.
- provide `isEqualTo` member function that determines whether two sets are equal.

Now write a driver program to test your `IntegerSet` class. Instantiate several `IntegerSet` objects. Test that all your member functions work properly.