



# Faculté des sciences appliquées

**FSA**

SINF1150 **Algorithmique et programmation**

[60h+60h exercices] 12 credits

**Teacher(s):** Baudouin Le Charlier (coord.), Marc Lobelle, Kim Mens, Peter Van Roy  
**Language:** french  
**Level:** 1st cycle course

## Aims

- \* To be aware that programming (and thus also computer science) are based on some very simple concepts which have almost not change since creation of computers by Von Neumann
  - \* To be aware that this activity is mainly based on reasoning and precision
- This point of view is totally different of that provided by medias and integrated by most people. They thinks that computer science is a set of moving techniques in perpetual revolution.

## Main themes

1. Introduction to algorithmic:
  - \* Two basic concepts: objects and actions
  - \* Approaches of algorithm composition
  - \* Decomposition of problems in sub-problems
  - \* Reasoning methods associated to the setup of algorithms (specifications, assertions, invariants)
2. Internal structure of computers
  - \* Basic knowledge of operational systems of computers
  - \* Structure of machine language
  - \* Principles of data representation
3. Main characteristics of the structure of high-level programming languages
  - \* Illustration through the Java language
  - \* Link with the underlying structure of computers
4. First introduction to object-oriented programming including structuring of classes and writing of graphical interfaces

## Content and teaching methods

### Methods:

The course is associated to

- \* exercises in order to induce a better assimilation of concepts,
- \* practical works realized in groups of 2 students to give students the opportunity to apply concepts.

### Content:

- \* Basic concepts: variable, expression, affectation; control structures: sequential, alternative and repetitive composition; simple data types: array, files, records; procedural abstraction and parameter transfer; classes and instances; program life cycle
- \* Methods to build a program: modeling and specification of a problem; downstream programming; inductive reasoning: induction on calculus, notion of invariant; informal demonstration of the validity of the program; tests; notion of complexity analysis; style description; application to the rigorous building of typical algorithms: research, sorting, etc.
- \* Data representation: binary system, integer in complement to two, floating-point numbers, coding of characters
- \* Internal structure of computers: central unit, peripheral units, machine language, programming in machine language
- \* Programming languages: simple data types, operation priority, overload of names,  $\iota$
- \* Object-oriented programming: classes defined by the user, graphical classes, programming of a simple graphical interface

**Other information (prerequisite, evaluation (assessment methods), course materials recommended readings, ...)**

\* Prerequisite: none

\* Evaluation :

Individual and written tests and exam. Three notes contribute to the final course note: practical works, test in January, and exam (in June or September).

**Other credits in programs**

<b>LINF1EP</b>	Année de formation préparatoire à la licence en informatique	(12 credits)	Mandatory
----------------	--	--------------	-----------