	Prerequisites:	No prerequisites
	Learning outcomes:	Students know and understand the principles of iteration and recursion, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), the basis of algorithmic analysis (sorting, searching). Able to design and write an algorithmic solution for a specific problem by using of the data structures, independently of a specific language.

№	Lecture	Hours	Labaratory works		Literature
			Contents	Hours	
Part I: Basic Control and Data Structures					
1.	**Notion of Algorithm, Sequential Organisation of Computation**				
	Introduction, notion of algorithm, notion of program, problem analysis, notion of information, notion of value, notion of operation, notion of action, notion of status, notion of algorithm, algorithmic notation, notion of computer and programs, sequential composition of actions, naming and typing, basic types (integer, real, boolean, character, string), reading data, writing results, example of a first algorithm	2 h	Writing simple algorithms doing simple computations without any case analysis or iteration, manipulation of the basic types and basic arithmetic and Boolean expressions	2 h	
2.	**Sequential Organisation of a Computation Case Analysis**				
	Other examples of sequential composition, manipulation of strings, Introduction to case analysis, notations (if then else, multiple cases), strategies of case analysis (cases definition according to the expected results, nested case analysis, serialized case analysis)	2 h	Exercises about case analysis needing to apply the different strategies presented in the course	2 h	
3.	**Structured information (structured types) Notion of function**				
	Structured information, structured named types (dates, times, etc.), corresponding notation (fields designation), notion of function, specification of a function, examples of functions, functions using structured types, usage of a function (call), definition of formal parameter and effective parameter.	2 h	Specification and writing functions and writing main algorithms calling functions.	2 h	
4.-5.	**Tabulated functions Description of the System States, Named actions**				
	Notion of table, example of a tabulated function (function represented into a table), Specification of named action (effect, initial state, final state), different types of parameters (read, elaborated, modified), global variable, local variable, an example of an action repository (the drawing machine of S. Papert), setting up the drawing machine and system states formal description. Management of the variables and the parameters of actions and functions: creation of the parameters and local variables, retrieving result of a function, static variables, life duration of local/global variables, of parameters.	4 h	Specification and writing of actions and functions, and exercises using the drawing machine, states description. Exercises about the management of the variables and parameters by the system during the execution of an algorithm.	4 h	

6.	**Introduction to iterative algorithms**				
	Iterative algorithms notations (repeat n times, while, repeat until), notion of sequence, enumeration of a sequence, research in a sequence, simple examples of iterative algorithms, invariant property of an iterative algorithm. Sequential files: reading files, files creation.	2 h	Writing of simple iterative algorithms	2 h	
7.	**Reasoning with abstract sequences**				
	General specification of a sequence, general algorithmic models for processing a sequence, general algorithmic models for searching a specific element of a sequence.	2 h	Writing iterative algorithms by applying the presented models.	2 h	
8.	**Different representations of a sequence of information**				
	Characterization of a sequence. Different ways of representing a sequence of information: files, one dimension arrays, strings, computed sequences. Example of a same algorithm adapted to different representations, Different sequential algorithms applied on a specific representation	2 h	Specification and realization of algorithms involving various sequences representations	2 h	
9.	**Sequential Processing of Arrays**				
	Sets, relations, queues, stacks represented by one dimension arrays, multi-dimension arrays	2 h	Application exercises with one or two dimensional arrays	2h	
10.	**Dynamic linear data structures**				
	Pointer and address, direct and indirect access, linked lists, associated memory management, sequential management of a linked list, building a linked list, adding, deleting elements of a linked list	2 h	Application exercises with linked lists	2 h	
11.	**Intermediate Sequences**				
	Definition of logical sequences built upon a physical sequence. Examples of intermediate sequences, Specification of an intermediate sequence	2 h	Specification and realization of algorithms involving various intermediate sequences	2 h	
12.	**Récursion**				
	Designing and writing recursive functions and procedures. Proof and evaluation of a recursive algorithms. Systematic transformation of a recursive algorithm to an iterative algorithm.	2 h	Specification and realization of recursive functions and procedures	2 h	
13-14.	**Trees & Graphs**				
	Binary trees. Basic Operations on Binary Trees. Tree Search and Insertion. Analysis of Tree Search and Insertion. Graphs (oriented and unoriented). Basic opertion with graphs. Methods of graph girth.	4 h	Specification and realization of algorithms manipulating trees	4 h	
15.	**Sorting algorithms**				
	Sorting by Straight Insertion, Sorting by Straight Selection, Sorting by Straight Exchange, Advanced Sorting Methods, Sorting Sequences	2 h		2 h	
	TOTAL:	**30 h**		**30 h**	
	Literature:				

[1] Introduction to Algorithms (Second Edition) by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein, published by MIT Press and McGraw-Hill.

[2] The Design and Analysis of Computer Algorithms by Alfred V. Aho, John E. Hopcroft and Jefrey D. Ullman, Addison-Wesley, 1974

[3] Algorithms and data structures by N.Wirth, Prentice Hall, New Jersey, 2001