



INF - Computer Science - S5-S6



ALGORITHMES ET STRUCTURES DE DONNÉES

ALGORITHMS AND DATA STRUCTURES

Lecturers: Romain VUILLEMOT

| Lecturers : 8.0 | TC : 17.0 | PW : 0.0 | Autonomy : 5.0 | Study : 0.0 | Project : 0.0 | Language : FR

Objectives

The objective of this course is to introduce the fundamentals of algorithms and data structures, necessary for students who intend to become engineers. Students will be introduced to the analysis of problems, the design and implementation of algorithms but also to their applications in the industry, through lectures, practical sessions and a business opening. practical work sessions and a business opening. The concepts covered will be implemented in the Python language.

Keywords : algorithmic, data structures, problem solving, algorithm implementation, complexity complexity

Programme

- Data structures.
- Introduction to complexity.
- Sorting algorithms.
- Graph algorithms.
- General paradigms and examples: divide and conquer, dynamic programming, gluttonous algorithms algorithms, heuristics.

Learning outcomes

Independent study

Objectifs : Understand and assimilate the course concepts implemented in the TDs.

Méthodes :

Core texts

T. H. Cormen, C. E. Leiserson, R. L. Rive, *INTRODUCTION TO ALGORITHMS*, The MIT Press and McGraw-Hill Book Company, 2001., 2009

Assessment



CONCEPTION ET PROGRAMMATION OBJET

OBJECT-ORIENTED DESIGN AND PROGRAMMING

Lecturers: Emmanuel DELLANDREA

| Lecturers : 8.0 | TC : 17.0 | PW : 0.0 | Autonomy : 5.0 | Study : 0.0 | Project : 0.0 | Language : FR

Objectives

The goal is to provide students with a basic knowledge of the design and development of programs using the object approach. These notions will be implemented through the Python language for programming and UML for modelling. The traditional lectures are limited in order to favour a practical approach in the form of programming exercises and short projects, some of which are evaluated.

Keywords : Object programming, object-oriented design, Python programming language

Programme

- Object-oriented design: Classes and instances. Encapsulation, attributes, methods. Aggregation, composition. Inheritance and polymorphism. Operator overloading.
- Implementing concepts in Python.
- Development of graphical interfaces.

Learning outcomes

- Know how to establish an object-oriented model of a computer application.
- Know how to use the UML formalism.
- Know how to implement an object model using the Python language.
- Know how to implement an IT project, and write a report.

Independent study

Objectifs : To understand and deepen the course concepts implemented during practical works.

Méthodes : Q&A sessions with teachers following the practical works to help with the completion of homework assignments.

Core texts

Bertrand Meyer, *CONCEPTION ET PROGRAMMATION ORIENTÉES OBJET.*, Eyrolles, 2008
Delannoy Claude *S'INITIER À LA PROGRAMMATION ET À L'ORIENTÉ OBJET.*, Eyrolles, 2016
Pascal Roques, Franck Vallée *UML 2 EN ACTION : DE L'ANALYSE DES BESOINS À LA CONCEPTION*, Eyrolles, 2007

Assessment

Grade = 50% knowledge + 50% know-how
Knowledge grade = 100% final exam
Know-how grade = 100 % Average of 2 reports



PROJET D'APPLICATION WEB

PRACTICAL IT PROJECT

Lecturers: Daniel MULLER, René CHALON

| Lecturers : 8.0 | TC : 10.0 | PW : 0.0 | Autonomy : 12.0 | Study : 0.0 | Project : 0.0 | Language : FR

Objectives

The objective of this lecture is to put into practice the notions and skills introduced by the other lectures of Computer Science, in the framework of a group project based on the design and development of a web application. The architecture of this application will be based on a dynamic web interface, and a service developed in Python on the server side.

The project will concern the whole application: data model, server, user interfaces (GUI) and client-server interactions.

Keywords : WebApp, Webservice, client-server architecture, HTML5, CSS3, Javascript, Ajax, JSON, Python, SQL.

Programme

- Web development context and tools.
- Data model.
- Client-server architecture and application protocol.
- Principle of a server.
- User interfaces.

Learning outcomes

- Master the basic tools and languages for web programming (HTML, CSS, Javascript).
- Understanding client-server architecture based on web services.
- Know how to do software testing.
- Master the specific characteristics of IT project management (versioning).

Independent study

Objectifs : Design, development and testing of an application

Méthodes : The students are divided into project groups of 4 to 5 people. They have to write a specification, deliver a functional and tested application, and perform the acceptance test of an application made by a competing group.

Core texts

Ben Frain, *RESPONSIVE WEB DESIGN WITH HTML5 AND CSS: DEVELOP FUTURE-PROOF RESPONSIVE WEBSITES USING THE LATEST HTML5 AND CSS TECHNIQUES, 3RD EDITION*, Packt, 2020

David Flanagan *JAVASCRIPT: MASTER THE WORLD'S MOST-USED PROGRAMMING LANGUAGE*, O'Reilly, 2020

Leonard Richardson *RESTFUL WEB APIS: SERVICES FOR A CHANGING WORLD*, O'Reilly,

Assessment

Final mark = 50% knowledge + 50% know-how (Knowledge = 100% final exam and know-how = 100% project deliverables).