

**Informatique S2**  
**Computer Science**

**Code cours** *Course code:* **INF2**

**Crédits ECTS** *ECTS Credits:* **3**

<b>Département</b> <i>Department</i>	: IA	<b>Cours Lectures</b>	: 7H30
<b>Coordonnateurs</b> <i>Lecturers</i>	: M. RICHARD	<b>T.D. Tutorials</b>	: 11H15
<b>Période</b> <i>Year of study</i>	: 1 <sup>ère</sup> année <i>1<sup>st</sup> year</i>	<b>T.P. Laboratory sessions</b>	: 6H
<b>Semestre</b> <i>Semester</i>	: 2 <sup>ème</sup> semestre <i>2<sup>nd</sup> semester</i>	<b>Projet</b> <i>Project</i>	: 7H30
<b>Evaluation</b> <i>Assessment method(s)</i>	: 1 examen écrit, 2 TP, 1 projet <i>1 written exam, 2 practical work tests, 1 project</i>	<b>Non encadré</b> <i>Homework</i>	: 7H30
<b>Langue d'instruction</b> <i>Language of instruction</i>	Français <i>French</i>	<b>Horaire global</b> <i>Total hours</i>	: 39H45
<b>Type de cours</b> <i>Type of course</i>	Obligatoire <i>Compulsory</i>		
<b>Niveau</b> <i>Level of course</i>	: débutant <i>Undergraduate</i>		

**Compétences attendues :**

À l'issue de ce module, l'étudiant devra être capable :

- de concevoir des unités modulaires fortement réutilisables et utilisable par faible couplage
- de spécifier concevoir et implémenter des algorithmes robustes
- de manipuler et mettre en œuvre (conception et implémentation) des structures dynamiques linéaires

**Pré-requis :** Informatique 1 (cours A1, semestre 1)

**Contenu :**

*Cours/TD/TP :* Ce module fait suite au module « Informatique » du premier semestre. Trois grands axes seront développés lors de celui-ci :

- la conception et réalisation d'unités modulaires réutilisables et faiblement couplées :
  - o il s'agit ici de maîtriser est mettre en pratique les concepts d'encapsulation et de généricité permettant de concevoir API et/ou modules fortement réutilisables, permettant ainsi de réduire les coûts de développement et d'améliorer drastiquement la qualité du code.
- l'écriture de spécifications formelles d'unités algorithmiques (sous-programmes) :
  - o les concepts d'assertions et de spécifications une fois présentés, nous verrons 2 grands schémas de spécifications : à priori et à posteriori.
  - o Le concept d'exception, indispensable à la bonne mise en œuvre du dernier schéma cité, sera également présenté et utilisé.
- les structures dynamiques linéaires :
  - o ce chapitre présente, dans un premier temps, la notion de pointeur typé. Cette notion sera alors utilisée pour la conception, la réalisation et l'utilisation de structures de données dynamiques linéaires (pile, file deque, ...)

*Projet :* Le but du projet de programmation de ce module est de réaliser l'analyse, la programmation en ADA et la mise au point d'une application de taille respectable. Les étudiants, répartis en équipe de 4 à 5, s'organiseront pour spécifier, concevoir et implémenter un logiciel robuste et de qualité répondant au cahier des charges fourni en début de projet.

**Bibliographie :**

**Expected competencies:**

Upon completion of this module, the student should be able to:

- design modular units that are highly reusable and usable with weak coupling
- specify, design and implement robust algorithms
- manipulate and implement (design and implement) linear dynamic structures

**Prerequisites:** course Computer science semester 1

**Content:**

Course/DD/PT: This module follows the "Computer Science" module of the first semester. Three main axes will be developed during this one :

- the design and realization of reusable and loosely coupled modular units:

o the aim is to master and put into practice the concepts of encapsulation and genericity allowing the design of APIs and/or highly reusable modules, thus reducing development costs and drastically improving code quality.

- writing formal specifications of algorithmic units (subprograms):

o Once the concepts of assertions and specifications have been presented, we will see 2 main specification schemes: a priori and a posteriori.

o The concept of exception, which is essential to the good implementation of the last mentioned scheme, will also be presented and used.

- linear dynamic structures:

o this chapter presents, in a first step, the notion of typed pointer. This notion will then be used for the design, the realization and the use of linear dynamic data structures (stack, queue, ...)

Project: The goal of the programming project of this module is to carry out the analysis, the programming in ADA and the development of an application of respectable size. The students, divided into teams of 4 to 5, will organize themselves to specify, design and implement a robust and quality software that meets the specifications provided at the beginning of the project.

**Recommended reading:**