

Types de Données Abstraites & Base de la Programmation Orienté Objet <i>Abstract Data Type & Base of Object-Oriented Programming</i>			
Code ECUE Course code: TAB		UE : UE5-1	
Département <i>Department</i>	IA	Cours <i>Lectures</i>	17h30
Coordonnateurs <i>Lecturers</i>	M. Richard	T.D. <i>Tutorials</i>	17h30
Période <i>Year of study</i>	3 ^{ème} année <i>3rd year</i>	T.P. <i>Laboratory sessions</i>	
Semestre <i>Semester</i>	S5	Projet <i>Project</i>	
Evaluation <i>Assessment method(s)</i>	1 écrit	Non encadré <i>Unsupervised</i>	
Langue d'instruction <i>Language of instruction</i>	Français	Horaire global <i>Total hours</i>	35h00
Type de cours <i>Type of course</i>	Obligatoire	Travail personnel <i>Homework</i>	15h00
Niveau <i>Level of course</i>	Second cycle universitaire		

Compétences attendues :

À l'issue de ce module, l'étudiant possédera :

- une maîtrise avancée du langage de programmation C (modularité, gestion mémoire, ...)
- la capacité d'implémenter et d'utiliser des structures de données dynamiques linéaires et hiérarchiques (Pile, File, Deque, Arbre, Graphe)
- une bonne connaissance des concepts de base de la programmation orientée objet (Classes, Associations, Encapsulation, Abstraction, ...)

Pré-requis : Une pratique d'un langage de programmation procédural.

Contenu :

Ce module est composé de trois grandes parties permettant l'acquisition des compétences ci-dessus :

1. Langage C :

Ce chapitre permettra l'acquisition de compétences avancées dans l'utilisation du langage C, en particulier concernant la modularité, la gestion mémoire et les bonnes pratiques pour la mise en place de d'algorithmes robustes

2. Types Abstraites de Données linéaires et hiérarchiques :

Nous étudierons ici les grandes familles de types abstraits de données dans un premier temps linéaire (Pile, file, listes, queue, deque, ...) puis hiérarchique (arbre, arbre binaire, arbre binaire de recherche, introduction aux graphes).

L'étude des algorithmes permettra une initiation à la complexité en espace et en temps de ces derniers.

3. Des types de données abstraits au paradigme objet :

De manière très naturelle nous découvrirons, à partir des connaissances acquises dans la partie précédente, le paradigme de la programmation objet et les éléments de base de ce type de programmation (classes, relation d'association, d'agrégation et composition, encapsulation, abstraction et héritage) permettant d'améliorer les propriétés d'un code selon différents critères définis en génie logiciel.

Toutes ces notions seront largement mises en œuvre au cours des différentes séances de travaux dirigés

Bibliographie :

Aucune