

FICHE ECTS

Composante (dépt ou centre)	DGEI
Code ECTS	I4IRIL11
Crédits ECTS	5
Enseignant responsable de l'UF	Sami YANGUI
Section CNU	
Libellé de l'UF (français)	Conception et Programmation Orientées Objet
Libellé de l'UF (anglais)	Object Oriented Design and Object Oriented Programming
Semestre :	Semestre 1 <input checked="" type="checkbox"/> Semestre 2 <input type="checkbox"/> Annuel <input type="checkbox"/>

DESCRIPTION GENERALE

VERSION FRANÇAISE

Objectifs, finalités ¹ (max 1000 caract.)

A la fin de ce module, l'étudiant devra avoir compris et pourra expliquer (principaux concepts) :
Conception orientée objet à travers le langage UML, et programmation orientée objet à travers le langage Java.

L'étudiant devra être capable de :

- * Maîtriser la théorie objets et le langage de modélisation UML
- * Maîtriser une méthodologie de conception logiciel guidée par les cas d'utilisation et comprenant les étapes d'analyse et de conception détaillée
- * Mettre en pratique les concepts objet et une méthodologie de conception dans le langage de programmation JAVA
- * Comprendre les avantages d'utilisation des bonnes pratiques de conception basées sur les designs patterns
- * Configurer et mettre en pratique des outils de gestion de configuration (svn, cvs, etc.)
- * S'initier à l'utilisation des documents standard pour la gestion de projet, spécification d'exigences, conception et tests
- * Configurer et mettre en œuvre et utiliser des espaces collaboratifs pour le processus de développement logiciel
- * S'initier aux problématiques de la gestion d'un projet de développement logiciel
- * Mettre en œuvre (planifier) et jouer les différents rôles de concepteur et développeur dans un processus de développement logiciel.

*Le verbe qui décrit l'objectif doit décrire de façon précise l'effet attendu, on conseille d'éviter des verbes vagues pour lesquels il sera difficile de vérifier si l'objectif est atteint (connaître, comprendre, apprendre, savoir...)
On peut s'inspirer des verbes d'actions de la taxonomie de Bloom*

VERSION ANGLAISE

Objectives (max 1000 charact.)

At the end of this module, the student will have understood and be able to explain (main concepts):
Object oriented based on the UML language, and the object oriented programming through based on the Java language.

The student will be able to:

- * Master object theory and the design modelling language UML
- * Master a design methodology based on use cases and integrating detailed analysis and design phases.
- * Apply the object concepts and a design methodology using the JAVA language
- * Understand the advantages of following best practises guidelines provided by the use of design patterns
- * Configure and use the configuration management tools (e.g. svn, cvs, etc.)
- * Explore the use of standard documents for project management, requirements specification, software design and software tests.
- * Configure and use collaborative workspaces applied to the software development process.
- * Have an initial experience to the project management challenges in a software development process project.
- * Plan and play designer and developer roles within a software development process.

Contenu (max 1000 caract.)	Description (max 1000 caract.)
Une première partie présente les concepts fondamentaux du modèle objet ainsi que la notation utilisée pour décrire les modèles de conception, à travers le langage UML. Une introduction à un processus de développement est donnée au fil des travaux dirigés, en illustrant les enchaînements entre les principaux diagrammes. Une deuxième partie permet la mise en œuvre de ces concepts dans un langage de programmation orientée objet : Java. Sont présentés en particulier le système de développement Java, les bases traditionnelles de la programmation en java ainsi que l'aspect objet, des exemples de l'API standard, le traitement des flux de données, la plateforme SWING et la programmation multi-threads.	A first part presents the fundamental concepts of the object model as well as the notation used to describe the models of design, through language UML. An introduction to a process of development is given during directed work (TD), by illustrating the sequences between the principal diagrams: case of use, classes, sequence, states machines. These TD also makes it possible to the students to be initiated with the an UML tool. A second part allows the implementation of these concepts in an object programming language: Java. Are presented in particular the system of development Java (JDK), the traditional bases of the programming in java as well as object aspects (classes, methods,...), examples from the standard API, treatment of the data flows (IO), the SWING framework and the multi-threads programming.
<i>Vous pouvez aussi mentionner ci-dessus les documents remis aux étudiants</i>	

Recommandation (max 1000 caract.)	Recommendation (max 1000 caract.)
<i>Principales difficultés habituellement rencontrées par les étudiants</i>	

Pré-requis (Code UF + intitulé, sinon notions nécessaires) (max 200 caract.)	Necessary knowledge (UF Code + title, or required knowledge) (max 200 caract.)
Connaissance de la programmation structurée (Ada, C, Pascal,...)	Structured programming (ADA, C, Pascal,...)

Organisation, méthodes pédagogiques			Organisation, teaching methods		
La formation se déroule en appliquant une méthode APP adapté au processus développement logiciel 1) Cours et TDs de conception. Les TDs de conception s'appuient sur un cas d'étude. 2) Cours et TDs de programmation. 3) TPs pour construire la solution conçue sur la base de la conception réalisée en 1) et les connaissances de programmation en 2)			The course follows a PBL based methodology adapted to software development process and called yPBL 1) Course and labs in designing applied to a case study 2) Course and labs in programming the same case study. 3) Labs aimed at building the designed study case and the explored technologies in 1) and 2) 4) Final project.		
Horaire présentiel (tel que l'enseignement est comptabilisé)		Travail personnel²	Contact hours		Personal work
CM	17,5		Lectures	17,5	
TD	25		Tutorials	25	
TP	33		Lab work	33	
Projet			Project		
Examen formatif			Coursework		
Examen certificatif	1,25		Exam	1,25	

Format d'enseignement :	Présentiel <input checked="" type="checkbox"/>	Distanciel <input type="checkbox"/>	Hybride <input type="checkbox"/>
--------------------------------	--	-------------------------------------	----------------------------------

DESCRIPTION COMPLEMENTAIRE

Modalités d'évaluation (max 1000 caract.)		Assessment (max 1000 charact.)	
Comment évaluez-vous que ces objectifs sont atteints ?			
Comment évaluez-vous que ces objectifs sont atteints ? La lecture et écriture des modèles objets UML et sa traduction dans le langage Java En quoi le système d'évaluation aide les étudiants à atteindre ces objectifs ? Renforcement des concepts objets par la modélisation et le développement.		How do you assess that the objectives have been reached? Reading and writing UML models and translating them in Java language How does your assessment system help the student to reach the objectives? Objects definitions learning reinforced by modelling and development.	
Examen écrit <input type="checkbox"/> oral <input type="checkbox"/> Rapport <input type="checkbox"/> Exposé <input type="checkbox"/> TP <input type="checkbox"/>		Written ex. <input type="checkbox"/> Oral ex. <input type="checkbox"/> Report <input type="checkbox"/> Presentation <input type="checkbox"/> Labwork <input type="checkbox"/>	
Autre (préciser)		Other (please describe)	

Aides aux étudiants	Student aid

Public ciblé	Student aid
Type de formation	Formation initiale <input checked="" type="checkbox"/> Formation continue <input type="checkbox"/> Apprentissage <input type="checkbox"/> VAE <input type="checkbox"/>

Admission	Admission

Besoins particuliers	Particular needs

Langue(s) utilisée(s) pour l'enseignement :	Français
Langue(s) utilisée(s) pour le support de cours :	Anglais
Langue(s) utilisée(s) pour l'évaluation :	Anglais

Mots clés :	Keywords :
COO, POO, UML, JAVA	OOD, OOP, UML, JAVA

Bibliographie (auteur, titre, éditeur, année, ISBN)	Bibliography (author, title, publisher, year, ISBN)
Bibliographie, webographie, polycopié... Il s'agit de documents accessibles aux étudiants pour mieux maîtriser votre enseignement	
•« Object-Oriented Analysis and Design: Understanding System Development with UML 2.0 » by Mike O'Docherty •« Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development » by Craig Larman •« Design Patterns: Elements of Reusable Object-Oriented Software » by Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995), Addison-Wesley,	

¹http://enseignants.insa-toulouse.fr/fr/ameliorer_mon_cours/comment_rediger_les_objectifs_de_son_enseignement.html

² à titre d'exemple, on peut multiplier le présentiel par un facteur fonction du type de pédagogie : 0,9 pour les CM, 0,7 pour les TD, 0,3 pour les TP, 1,5 pour les APP et autres pédagogies actives. Dans tous les cas, cette valeur doit être la plus authentique possible et s'appuyer sur des moyens appropriés pour guider le travail personnel de l'étudiant (exercices non corrigés lors des TD, préparation de TP, exercices « pour aller plus loin », grilles d'auto-évaluation, travail personnel à faire utilisant la bibliographie recommandée, ...)