QF01/0408-4.0E	Course Plan for Master program - Study Plan Development and Updating Procedures/ Department

Study plan No.	**2021-2022**		**University Specialization**		**Software Engineering**	
Course No.	**0104754**		**Course name**		**Advanced Software Modeling**	
Credit Hours	**3**		**Prerequisite Co-requisite**		**-**	
Course type	☐ MANDATORY UNIVERSITY REQUIREMENT	☐ UNIVERSITY ELECTIVE REQUIREMENTS	☐ FACULTY MANDATORY REQUIREMENT	☐ Support course family requirements	☐ Mandatory requirements	☐ Elective requirements
Teaching style	☐ **Full online learning**		☐ **Blended learning**		☒ **Traditional learning**	
Teaching model	☐ **2Synchronous: 1asynchronous**		☐ **2 face to face : 1synchronous**		☒ **3 Traditional**	

Faculty member and study divisions information (to be filled in each semester by the subject instructor)

Name	Academic rank	Office No.	Phone No.	E-mail	
Division number	**Time**	**Place**	**Number of students**	**Teaching style**	**Approved model**

Brief description

This course provides an advanced view of the software Models. The student in this course will learn how to create, built, drew, test and refactor software models; specifically UML models. The course aims to give the students real challenges so they can put their modeling knowledge into practice.

Learning resources

Course book information (Title, author, date of issue, publisher ... etc)	Rumpe, B. (2017). Agile Modeling with UML: Code Generation, Testing, Refactoring. Springer.			
Supportive learning resources (Books, databases, periodicals, software, applications, others)	**1.** Rumpe, B. (2016). Modeling with UML: Language, Concepts, Methods (1st ed. 2016 ed.). Springer. 2. Douglass, B. P. (2021). Agile Model-Based Systems Engineering Cookbook: Improve system development by applying proven recipes for effective agile systems engineering. Packt Publishing. 3. Borky, J. M., & Bradley, T. H. (2019). Effective Model-Based Systems Engineering (1st ed. 2019 ed.). Springer.			
Supporting websites				
The physical environment for teaching	☒ Class room	☐ labs	☒ Virtual educational platform	☐ Others
Necessary equipment and				

QF01/0408-4.0E	Course Plan for Master program - Study Plan Development and Updating Procedures/ Department

software	
Supporting people with special needs	
For technical support	

Course learning outcomes (S= *Skills, C= Competences K= Knowledge,)*

No.	Course learning outcomes	The associated program learning output code
	Knowledge	
K1	A student will be able to understand and discuss the need, goals and tasks for software Models.	**MK2**
K2	A student will understand different software models.	**MK2**
K3	A student will understand the process of creating and applying software models	**MK3**
	Skills	
S1	A student will be able to create a UML models	**MS2**
S2	A student will be able to Generate code using UML models	**MS2**
S3	A student will be able execute and validate UML models	**MS2**
	Competences	
C1	A student will have the ability to create, execute and validate UML models.	**MC3**
C2	A student will have the ability to extract code using UML models	**MC3**

Mechanisms for direct evaluation of learning outcomes

Type of assessment / learning style	Fully electronic learning	Blended learning	Traditional Learning (Theory Learning)	Traditional Learning (Practical Learning)
First exam	0	0	%20	**0**
Second / midterm exam	%30	%30	%20	**30%**
Participation / practical applications	0	0	10	**30%**
Asynchronous interactive activities	%30	%30	0	**0**
final exam	%40	%40	%50	**40%**

Note: Asynchronous interactive activities are activities, tasks, projects, assignments, research, studies, projects, work within student groups ... etc, which the student carries out on his own, through the virtual platform without a direct encounter with the subject teacher.

Schedule of simultaneous / face-to-face encounters and their topics

Week	Subject	learning style*	Reference **
1	Introduction	Lecture	Pages (textbook) 6
2	Agile and UML-Based Methodology	Lecture	Pages (textbook)

			9
3	Compact Overview of UML/P	Lecture	Pages (textbook) 33
4	Object Constraint Language	Lecture	Pages (textbook) 39
5	Principles of Code Generation	Lecture	Pages (textbook) 71
6	Code Generation Techniques	Lecture	Pages (textbook) 82
7	Semantics of Code Generation	Lecture	Pages (textbook) 89
8	Transformations for Class Diagrams	learning through projects	Pages (textbook) 100
9	Transformations for Object Diagrams	learning through projects	Pages (textbook) 123
10	Code Generation from OCL	learning through projects	Pages (textbook) 132
11	Executing State charts	Lecture	Pages (textbook) 146
12	Review of previous chapters Midterm Exam (30 %)		
13	Principles of Testing with Models	Lecture	Pages (textbook) 161
14	Model-Based Tests	learning through projects	Pages (textbook) 185
15	Design Patterns for Testing	Lecture	Pages (textbook) 99
16	Final Exam		

*** Learning styles: Lecture, flipped learning, learning through projects, learning through problem solving, participatory learning ... etc.**

**** Reference: Pages in a book, database, recorded lecture, content on the e-learning platform, video, website ... etc.**