POLITÉCNICA

UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

(Subject Name)

Learning Guide – Information for Students

1. Description

Grade	**European Master on Software Engineering**
Module	Software Development
Area	
Subject	Software Architecture
Type	Mandatory
ECTS credits	4
Responsible department	DLSIIS
Major/Section/	

Academic year	2012/2013
Term	2nd term
Language	English
Web site	

2. Faculty

NAME and SURNAME	OFFICE	email
Jaime Ramírez Rodríguez (Coord.)	5112	jramirez@fi.upm.es

3. Prior knowledge required to take the subject

Passed subjects	
Other required learning outcomes	• Object oriented design

4. Learning goals

SUBJECT-SPECIFIC COMPETENCES AND PROFICIENCY LEVEL		
Code	Competence	Level
SC12	To conceive and perform the design of software systems, assuring relevant quality attributes.	A

Proficiency level: knowledge (K), comprehension (C), application (A), and analysis and synthesis (S)

SUBJECT LEARNING OUTCOMES			
Code	**Learning outcome**	**Related competences**	**Proficiency level**
LR1	To design the system according to the requirements, constraints, quality norms and organization goals.	SC12	A
LR2	To apply the architectural concepts that are relevant in the architectural design	SC12	A
LR3	Ability to document the software architecture	SC12	A

5. Subject assessment system

ACHIEVEMENT INDICATORS		
Ref	Indicator	Related to LR
I1	Distinguish what a software architecture is and what it is not	LR1, LR2
I2	Identify and priorize the quality attributes to be considered in the architecture design	LR1, LR2
I3	Understand the influence of the stakeholders and the organization in the architecture design	LR1, LR2
I4	Define quality attributes scenarios for the target architecture	LR2
I5	Know the main architecture patterns and styles	LR2
I6	Use suitable tactics and architecture patterns for achieving the quality attributes specified for the target architecture	LR2
I7	Know the basis of the product lines	LR2
I8	Know the available views for documenting an architecture	LR3

CONTINUOUS ASSESSMENT

Brief description of assessable activities	Time	Place	Weight in grade
Practical exercises on topics that are being explained in classroom	Weeks 3-12	Classroom	35%
Exam	Week 16	Classroom	25%
Research work on advance topics or real applications	Weeks 8-15	Out of classroom	30%
Oral presentation on the research works	Week 15	Classroom	10%
			Total: 100%

GRADING CRITERIA

Along the semester, in order to pass the course, the student will have to do the following exercises and works:

- Practical exercises: the student will have to do some practical exercises where he/she will have to apply the concepts, techniques and principles explained in the classroom.
- Final exam: the student will have to do a final exam where he/she will show that he/she has acquired the basic concepts explained in the classroom.
- Research work: the student will have to elaborate a state of the art on some advance topic or to analyze some real application from an architectural perspective. This work will have to give place to a document that will be delivered to the professor. In addition, prior to delivering this document, the student will have to do an oral presentation in classroom summarizing the preliminary results of his/her work.

The final mark (FM) will be calculated from the practical exercises mark (PEM), the exam mark (PM) and research work mark (RWM) by means of the following formula:

$$FM = 0.35*PEM + 0.25*EM + 0.4*RWM \quad \text{if } PEM \geq 4 \text{ and } EM \geq 4 \text{ and } RWM \geq 5$$

$$FM = 0 \quad \text{otherwise}$$

Where all the marks takes value between 0 and 10

When failed, in the extra exam period the final mark will be obtained from the mark of a research work and the mark of an exam. The minimum mark required in this period for both parts will be 4.

6. Contents and learning activities

SPECIFIC CONTENTS		
Unit / Topic / Chapter	**Section**	**Related indicators**
Chapter 1: Previous concepts on Software Architecture	1.1 What is software architecture?	I1
	1.2 The importance of a good software architecture	I2, I3, I8
Chapter 2: Defining a Software Architecture	2.1 Quality attributes related to software architecture	I2, I4
	2.2 Achieving quality attributes through tactics	I6
	2.3 Architectural styles	I5, I6
	2.4 Architectural patterns	I5, I6
	2.5 Product lines	I7, I8

7. Brief description of organizational modalities and teaching methods

TEACHING ORGANIZATION		
Scenario	**Organizational Modality**	**Purpose**
	Theory Classes	*Talk to students*
	Seminars/Workshops	*Construct knowledge through student interaction and activity*
	Practical Classes	*Show students what to do*
	Placements	*Round out student training in a professional setting*
	Personal Tutoring	*Give students personalized attention*
	Group Work	*Get students to learn from each other*
	Independent Work	*Develop self-learning ability*

TEACHING METHODS			
	Method	**Purpose**	
	Explanation/Lecture	***Transfer information and activate student cognitive processes***	Known as explanation, this teaching method involves the "*presentation of a logically structured topic with the aim of providing information organized according to criteria suited for the purpose"*. This methodology, also known as *lecture*, mainly focuses on the verbal exposition by the teacher of contents on the subject under study. The term *master class* is often used to refer to a special type of lecture taught by a professor on special occasions
	Case Studies	***Learning by analyzing real or simulated case studies***	Intensive and exhaustive analysis of a real fact, problem or event for the purpose of understanding, interpreting or solving the problem, generating hypotheses, comparing data, thinking, learning or diagnosis and, sometimes, training in possible alternative problem-solving procedures.
	Exercises and Problem Solving	***Exercise, test and practice prior knowledge***	Situations where students are asked to develop the suitable or correct solutions by exercising routines, applying formulae or running algorithms, applying information processing procedures and interpreting the results. It is often used to supplement lectures.
	Problem-Based Learning (PBL)	***Develop active learning through problem solving***	Teaching and learning method whose starting point is a problem, designed by the teacher, that the student has to solve to develop a number of previously defined competences.
	Project-Oriented Learning (POL)	***Complete a problem-solving project applying acquired skills and knowledge***	Teaching and learning method where have a set time to develop a project to solve a problem or perform a task by planning, designing and completing a series of activities. The whole thing is based on developing and applying what they have learned and making effective use of resources.
	Cooperative Learning	***Develop active and meaningful learning through cooperation***	Interactive approach to the organization of classroom work where students are responsible for their own and their peers' learning as part of a co-responsibility strategy for achieving group goals and incentives. This is both one of a number of methods for use and an overall teaching approach, or philosophy.
	Learning Contract	***Develop independent learning***	An agreement between the teacher and student on the achievement of learning outcomes through an independent work proposal, supervised by the teacher, and to be accomplished within a set period. The essential points of a learning contract are that it is a written agreement, stating required work and reward, requiring personal involvement and having a time frame for accomplishment.

BRIEF DESCRIPTION OF THE ORGANIZATIONAL MODALITIES AND TEACHING METHODS	
THEORY CLASSES	Some theoretical lectures will be arranged during the course to present basic concepts and key aspects, always supported by audiovisual resources and innovative techniques to enhance student comprehension
PROBLEM-SOLVING CLASSES	Professor and students will solve problems in the classroom to apply and fix the knowledge acquired during the theory classes
PRACTICAL WORK	Group work is complemented with practical classes in which students have to deal with problems and challenges as close as possible to real life developments
INDIVIDUAL WORK	Individual work will be intended for the study of the subject.
GROUP WORK	One kind of group work will be posed to the students: a research project.
PERSONAL TUTORING	Students will be able to attend personal tutoring, following the procedure established at the School

8. Teaching resources

TEACHING RESOURCES	
RECOMMENDED READING	Bass, L. et al. (2003) Software Architecture in Practice. Addison-Wesley, Boston, MA, second edition
	Buschmann, F. et al. (1996) Pattern-Oriented Software Architecture: A System of Patterns, volume 1 de Software Design Patterns. John Wiley & Sons.
	Taylor, R. N. et al. (2009) Software Architecture: Foundations, Theory and Practice. John Wiley & Sons.
	Bachmann, F. et al. (2007) Modificability Tactics. Inf. Téc. CMU/SEI-2007-TR-002, Software Engineering Institute - Carnegie Mellon University, Pittsburg, PA, USA.
	Gorton I. (2006) Essential Software Architecture. Springer-Verlag.
	Parnas, D. L. (1972) On the Criteria To Be Used in Decomposing Systems into Modules. Communications of the ACM, 15(12): págs. 1053-1058.
WEB RESOURCES	Subject Moodle site (http://)
EQUIPMENT	Room XXXX

9. Subject schedule

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
Week 1 (2 hours)	• Presentation					
Week 2-3 (6 hours)	• 1.1 What is software architecture? • 1.2 The importance of a good software architecture (4 hours)		• Study of the subject (2 hours)			
Week 4 (6 hours)	• 2.1 Quality attributes related to software architecture (2 hours)		• Study of the subject (1 hours)	• Exercise (3 hours)		
Week 5 (3 hours)	• 2.2 Achieving quality attributes through tactics (2 hours)		• Study of the subject (1 hours)			
Weeks 6-8 (10 hours)	• 2.3 Architectural styles (6 hours)		• Study of the subject (4 hours)			

Weeks 9-13 (50 hours)	• 2.4 Architectural patterns (10 hours)		• Study of the subject (5 hours)	• Exercise (23 hours) • Research work (12 hours)		
Week 14 (11 hours)	• 2.5 Product lines (2 hours)		• Study of the subject (2 hours)	• Research work (4 hours) • Preparation of oral presentations (3 hours)		
Weeks 15 (13 hours)	• Oral Presentations (2 hours)		• Study of the subject (4 hours)	• Preparation of oral presentations (3 hours)		
Week 16 (7 hours)	• Exam (2 hour)		• Study of the subject (5 hours)			
108 hours	• (32 hours)		• Study of the subject (24 hours)	• Research work (20 hours) • Exercises (26 hours) • Preparation of oral presentations (6 hours)		

Note: Student workload specified for each activity in hours