Centre	**University College of Engineering of Vitoria-Gasteiz**
Name of subject	**26017 – Software Engineering**
Qualification	**Degree in Computer Management and Information Systems Engineering**
Type	**Compulsory**
Credits	**6 ECTS**
Year	**2**
Term(s)	**2nd**
Department	**Computer Languages and Systems**
Language	**Spanish**

Outcomes / Objectives

Specific:

C1: Be able to distinguish the various stages of a software engineering process.

C2: Be able to understand an object-oriented software system in the UML language.

C3: Be able to design a software system applying multi-level architecture based on analysis. C4: Be able to implement a system based on the application design.

Cross-curricular:

C9b: Be able to communicate and convey knowledge, abilities and skills of the computer engineering profession.

Syllabus

UT1: Introduction to software engineering

- Motivation and software life cycle

- Objectives, properties and associated programming technologies.

UT2: Specification of UML artefacts

- Study of the different artefacts in UML

UT3: Multi-level software architectures: Presentation, Business Logic and Data

- Design of the different layers of a software system

UT4: Object-oriented design and programming

- Design of functionality

UT5: Implementation of a specific product

- Implementation of a software system using a set of current languages and tools

Methodology

Teaching Method

Face-to-Face Teaching Hours								
Lectures	Seminars	Classroom practice	Lab. practice	Computer sessions	Clinical practice	Workshops	Industrial workshops	Field practice
40			20					

Student Hours of Non Face-To-Face Activities								
Lectures	Seminars	Classroom practice	Lab. practice	Computer sessions	Clinical practice	Workshops	Industrial workshops	Field practice
40			50					

As this is a predominantly practical subject, LECTURES (M) will be used for presenting the theoretical concepts which will be subsequently used in the practical lessons, and for solving any queries students may have. Similarly, the concepts acquired in class will be reinforced through exercises, completed either individually or in small groups. Students may self-assess and correct the exercises completed during the subject with the use of the Web-CAT tool, which will provide them with an automatic assessment of the syntax and functionality of their implementations.

In the COMPUTER PRACTICE (GO), students will carry out a project that should realistically encompass the contents of the subject and will be done over the whole term. The project will be undertaken in groups and students must follow the jigsaw strategy, in such a way that they must first perform an individual task to analyse the proposed project, brainstorm alternatives, divide the tasks among the group members, form expert groups to discuss the solutions adopted, and integrate the results into the final product, which will be subsequently defended in an oral presentation.

MORE INFORMATION: - Moodle site of UPV/EHU for the subject Software Engineering.

Assessment System

General criteria

Written essay exam

Practical activities (exercises, case studies or problems)

Group assignments

Presentation of assignments, reading...

Clarification regarding assessment

Written exam(s) (60% of the final grade):

In the extraordinary examination session, students will sit a singe exam encompassing both parts. To pass the subject, students require a minimum grade of 3.5 in the final grade of the exam.

Deliverables (40% of the final grade)

Computer project (90%): The laboratory project will be carried out in a group following the jigsaw methodology. Any member of the group may be asked to explain any part of the project during the oral presentation of the project, and the grade obtained by that member will be applied to all the members of the group.

Other deliverables (10%): Reports on other design and programming tasks, subject forum, blog...

GRADE ON GRADE REPORT: Students may only opt for a single exam accounting for 100% of final grade in exceptional cases which have been justified at the beginning of the academic year and described in article 43 of the current student assessment regulations. Any exceptional cases must be notified to the lecturer prior to commencement of the subject or when the exceptional circumstance occurs, if it occurs throughout the course of the subject. No exceptional cases will be accepted past this time. Justification must be duly documented. In all other cases, written exams will always account for 60% of the final grade. The grade obtained for the practice will be saved throughout the year, but not for future academic years. If a student sits at least 2 exams in the ORDINARY examination session, the grade obtained will be the sum of both exams, otherwise it will be regarded as "Not Sat". If a student does not sit the exam in the EXTRAORDINARY examination session (June), it will be regarded as "Not Sat". Provided that the student sits the exam, the grade obtained from the practice in that year will be added to the grade obtained in the written exam.

Bibliography

Basic Bibliography

Construcción de Software Orientado a Objetos. Bertrand Meyer. Prentice-Hall. 1998.

In-depth Bibliography

Ingeniería del Software. Un enfoque práctico. Roger S. Pressman. MacGraw-Hill, 2001. 5th Edition.

Ingeniería El Proceso Unificado de Desarrollo de Software Jacobson, Booch, Rumbaugh. Addison Wesley, 1999

Websites

➢ http://jtf.acm.org/
➢ http://download.oracle.com/javase/tutorial/
➢ http://jga.sourceforge.net/