



Universitat de Lleida

DEGREE CURRICULUM  
**ENGINYERIA DEL  
PROGRAMARI**

Academic year 2013-14

## Subject's general information

<b>Subject name</b>	ENGINYERIA DEL PROGRAMARI
<b>Code</b>	102018
<b>Semester</b>	1r Q Avaluació Continuada
<b>Typology</b>	Obligatòria
<b>ECTS credits</b>	6
<b>Theoretical credits</b>	0
<b>Practical credits</b>	0
<b>Department</b>	Informàtica i Enginyeria Industrial
<b>Important information on data processing</b>	Consult <a href="#">this link</a> for more information.
<b>Language</b>	Català 60% Castellà 30% Anglès 10%
<b>Distribution of credits</b>	Juan Manuel Gimeno Illa 4.2 Montserrat Sendin Veloso 4.2

Juan Manuel Gimeno Illa  
Montserrat Sendin Veloso

## Subject's extra information

Compulsory subject of 3rd year (1st quarter) that belongs to the common studies in the computer science branch.

Matter: Analysis and Design of Applications.

**RECOMMENDATIONS:** We assume the student knows the concepts about Object-Oriented Programming taught in Programming II and Data Structures.

## Learning objectives

See section of competences.

## Competences

### Degree-specific competences

- **Knowledge and application of the principles, methodologies and life cycles of software engineering.**

#### Goals

- Apply the Use Case technique
- Developing the analysis classes diagram following the Object Oriented Modeling principles
- Developing the contracts of the operations
- Be familiar with a UML-based modeling tool
- Knowing the software lifecycle process models used over the years
- Understanding the development philosophy used in the Unified Process
- Be able to program basic unit tests
- Understanding and applying the object oriented design fundamental principles

- **Ability to plan, conceive, unfold and direct projects, services and computer systems in all fields, and manage their set up and continual improvement as well as value their economical and social impact.**

#### Goals

- Understanding the concept of code as a something that evolves over time
- Recognizing the concept of responsibility as a fundamental one when planning an object oriented design
- Knowing the conceptual basis and the different aspects of the discipline

### Degree-transversal competences

- **Ability to understand the user's needs expressed in a non-technical language.**

#### Goals

- Specifying in a textual way the functional needs of a certain software system planned by means of a statement and/or other inputs from the user
- Specifying in a textual way the non functional requirements of a certain software system planned by means of a statement and/or other inputs from the user
- Expressing graphically the flow of events, as a set of inputs and outputs, that describes the system behavior

## Subject contents

### **Theme I - *Introductory aspects***

- Initial questions about the Software Engineering
- A little of history
- Software development process
- Software process models
- Conclusions

### **Theme II - *Specifications and Requirements***

- Requirements Analysis
  - Initial concepts
  - Types of requirements
  - Examples
- The Use Cases technique
  - Concepts and components
  - Identification and specification of use cases
  - Examples
- A step more in the specification: the System Sequence Diagram
- Conclusions

### **Theme III - *Domain Analysis***

- Analysis Classes Diagram
  - Conception
  - Foundations of the Object Oriented Modeling
  - Domain Model construction
  - Examples
- A step more in the analysis: the contracts of the operations
- Conclusions

### **Theme IV - *Introduction to design and unit testing***

- Code as something which evolves
  - The concept of rotten code
  - Symptoms of rotten code
- The JUnit framework
  - Tests as executable specifications
  - Tests as facilitators of change
  - Testable code and flexible code

## Tema V - *The SOLID principles*

- Single responsibility principle
- Open-closed principle
- Liskov substitution principle
- Interface segregation principle
- Dependency inversion principle

## Tema VI - *Responsibility based design*

- The concept of responsibility
- The GRASP patterns of responsibility assignment

## Methodology

### PART PRESENCIAL

#### Grups Grans

##### • Classes Teoria (3 crèdits)

- Classes suportades amb transparències i/o apunts
- Es treballa sempre amb exemples

#### Grups Mitjans

##### • Classes Laboratori (3 crèdits)

- Us de l'eina de Modelat en UML: Visual Paradigm
- Treball continuat al voltant d'un cert **enunciat de pràctica**, que simularà el desenvolupament d'un projecte de software

### PART NO PRESENCIAL

- La pràctica es completarà en hores **No Presencials**
- **Es recomana** que l'alumne resolgui per compte propi els problemes de la **col·lecció de problemes**, a fi d'obtenir feedback per part del professor

## Evaluation

### Avaluació continuada

##### • 50% Teoria

- **Parcial 1:** 25%
- **Parcial 2:** 25%
- Si Parcial 1 ó Parcial 2 < 4 = Recuperació de la/es part/s corresponent/s
- Tipologia d'examen: fixació de conceptes i resolució de problemes

##### • 50% Pràctica

- Treball en parelles (Anàlisi) / individualment (Disseny)
- Entregues programades, dates no prorrogables
  - Anàlisi:

- Anàlisi de Requeriments: 15%
  - Pre-entrega: Diagrama de Casos d'Ús
- Model del Domini: 15%
- Disseny:
  - Proves unitàries: 20%

## • Requisites:

- **Mínim de 4** en cada part teòrica per ponderar amb la nota de pràctiques
- **Aprovat** = Nota Final  $\geq 5$

## Bibliography

### BASIC BIBLIOGRAPHY

#### - Introductory Subjects:

- I. Sommerville: "Ingeniería de Software".  
Prentice-Hall, 2005 (7ª ed.)

#### - Requirements:

- G. Kotonya, I. Sommerville: "Requirements Engineering: Processes and Techniques".  
Wiley, 1998
- A. Sutcliffe: "User-Centred Requirements Engineering. Theory and Practice".  
Springer-Verlag, 2002

#### - Unit Testing:

- P. Tahchiev et al.: Junit in Action (2nd edition). Manning, 2011.

#### - Object Oriented Methodology:

- C. Larman: "Applying UML and Patterns: An Introduction to Object- Oriented Analysis and Design and Iterative Development"  
Prentice-Hall, 2005 (3ª ed.)  
Spanish version: "UML y Patrones" \_ Prentice-Hall, 2002 (2ª ed.)
- Robert C. Martin: "Agile Software Development: Principles, Patterns, and Practices", Prentice-Hall, 2002.
- G. Booch, J. Rumbaugh, I. Jacobson: \_"El Lenguaje Unificado de Modelado".\_  
Addison-Wesley, 2006 (2ª ed.)
- J. Rumbaugh, I. Jacobson, G. Booch: "El Lenguaje Unificado de Modelado. Manual de referencia".\_  
Addison-Wesley, 2000
- I. Jacobson, G. Booch, J. Rumbaugh: \_ "El Proceso Unificado de Desarrollo de Software".\_  
Addison-Wesley, 2000

## COMPLEMENTARY BIBLIOGRAPHY

### - Introductory Subjects:

- R. S. Pressman: "Ingeniería de Software: Un enfoque práctico".  
McGraw-Hill, 2005 (6ª ed.)

### - Requirements:

- D. Kulak, E. Guiney: "Use Cases, Requirements in Context".  
Addison Wesley, 2000
- I. Jacobson: "Object-Oriented Software Engineering. A Use Case Driven Approach".  
Addison-Wesley, 1992

### - Object-Oriented Methodology:

- M. Fowler: "Refactoring: Improving the Design of Existing Code", Addison-Wesley, 1999.
- M. Fowler, K. Scout: "UML Gota a Gota".  
Addison-Wesley, 1999
- J. Conallen: "Building Web Applications with UML". Addison Wesley, 1999