



UNIVERSITAT D'ANDORRA

Seminar's teaching plan

Teaching plan	Bachelor in Computer Science
----------------------	-------------------------------------

Seminar	Software Engineering
Semester	4
Professor	Gerard Albà Soler galba@uda.ad
Mode	Face-to-face and virtual
Teaching language	English

1. Seminar presentation

The seminar covers all the fundamental topics for the development of quality software, with emphasis on flexible, reusable and maintainable software.

The first module of the course is an introduction to the main processes and methods of the lifecycle of software development. There is a specific section about the requirements process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed.

In the next modules, we address the subject of object-oriented analysis and design (OOA/D). Objects will be a key concept in the modeling of software. We study classes and objects, and their fundamentals of abstraction, encapsulation, inheritance and polymorphism.

We do a functional, dynamic and static modeling of the system or domain of our real problem using Unified Model Language (UML) notation. UML diagrams are used: use case, class, object, interaction and state machine diagrams, among others.

The final section discusses the concept and application of design patterns, a general repeatable solution to a commonly occurring problems in software OO design .

2. Seminar contents

Introduction to Software Engineering

1. Introduction
 - 1.1. What's Software Engineering
 - 1.2. Software Quality
 - 1.3. The Software Process (SLCD)
 - 1.4. Modeling techniques and paradigms
 - 1.5. Software Requirements
2. Object Oriented Fundamentals
 - 2.1. Introduction
 - 2.2. Object Oriented Thinking. Objects, Attributes, Classes and Operations
 - 2.3. Object Oriented Fundamentals. Abstraction, Encapsulation, Inheritance and Polymorphism
3. Object-Oriented Analysis and Design, and Functional Model
 - 3.1. What will we learn
 - 3.2. Object Oriented Analysis and Design
 - 3.3. OOA/D in the SDLC process
 - 3.4. Functional, Static and Dynamic views vs OOA/D
 - 3.5. The Unified Modeling Language (UML)
 - 3.6. Views of the modeling
 - 3.7. Diagrams
 - 3.8. Functional Model
4. Object Oriented Analysis: Dynamic Model

- 4.1. Use cases: actors, scenarios and textual description
- 4.2. UML Use case diagram
- 4.3. Graphical representation of use case scenario. UML Activity Diagram and UML System Sequence Diagram
- 4.4. System Operations and Operation Contracts
- 5. Object Oriented Analysis: Static Model
 - 5.1. From Requirements to Classes
 - 5.2. Conceptual model: UML Class diagram
 - 5.3. UML Object diagram
 - 5.4. UML Component diagram
- 6. Object Oriented Design
 - 6.1. Software Architecture
 - 6.2. Design principles
 - 6.3. Operation contract and Class diagram normalization
 - 6.4. Object responsibilities assignment
 - 6.5. Behavioral modeling. Sequence diagram and State diagram
 - 6.6. Design patterns

3. Seminar activities

3.1. Continuous assessment

Students must realize two face-to-face exams (CP) during the course. Additionally, two practical homeworks (TV) and a *challenge* (“el repte”) of the module are evaluated based on the following 3 learning outcomes.

The marks and results of each of the learning outcomes will be assessed based on the following weighting:

	TV 1	TV 2	CP 1	CP 2	CHALLENGE Mandatory	Total Evaluation
BInfo-E001-15	30%	0%	50%	0%	20%	100%
BInfo-E001-16	15%	35%	10%	40%	0%	100%
BInfo-E001-17	0%	10%	0%	30%	60%	100%

3.2. Final assessment

In this case, the final assessment contains 2 “treballs virtuals” (TV), 1 final exam (EF) and the challenge (“el reptre”) of the module evaluated through the following 3 learning outcomes:

	TV 1	TV 2	EF	CHALLENGE Mandatory	Total Evaluation
BInfo-E001-15	30%	0%	50%	20%	100%
BInfo-E001-16	15%	35%	50%	0%	100%
BInfo-E001-17	0%	10%	30%	60%	100%

4. Seminar resources

Basic bibliography

Professor material

Presentations and Case studies

Reference books

R Hyde (2020). *Engineering Software*. Ed no starch press

Booch G et al. (2007). 3rd Edition. *Object-Oriented Analysis and Design with Applications*. Ed Addison-Wesley Object Technology Series

Larman C. (2004). 3rd Edition. *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Ed Prentice Hall

Miles R. (2006). *UML 2.0*. Ed O'Reilly

Phillips D et al. (2018). 3rd Edition. *Python 3 Object-oriented Programming*. Ed Packt

Pressman R. et al. (2014). *Software Engineering. A Practitioner's approach*. 8th Edition. Ed McGrawHill

Roques P. (2004). *UML in Practice. The Art of Modeling Software Systems Demonstrated through Worked Examples and Solutions*. Ed Wiley

Sommerville I. (2015). *Software Engineering*. 10th Edition. Ed Addison-Wesley

Additional bibliography

McLaughlin B. (2007). *Head First Object Oriented Analysis and Design*. Ed O'Reilly

Weisfeld M. (2019). *The Object-Oriented Thought Process*. 3rd Edition. Ed Addison-Wesley Professional

Winter T. (2020). *Software Engineering at Google*. Ed O'Reilly

5. Seminar remarks

- Controls or exams are only repeated in exceptional cases. In order to be able to opt for the repetition of a control, the reason for the absence must be justified by presenting a supporting document.
- Late submissions will not be accepted.
- If a student does not submit any of the evaluable activities, the grade for the learning outcomes associated with the activity will be equal to zero.
- All evaluable submissions will be made on the UdA campus and in the format indicated in the statement of the activity.



Els continguts d'aquesta obra estan subjectes a una llicència de Reconeixement - No comercial - Sense obres derivades 3.0 de Creative Commons. Se'n permet la reproducció, distribució i comunicació pública sempre que se'n citi l'autor i no se'n faci un ús comercial. La llicència completa es pot consultar a: <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode>.