

SUBJECT PROGRAM

YO. SUBJECT IDENTIFICATION

Subject: Analysis and Design of Software		Acronym: INF-236	Approval date 10/11/2016 (CC.DD. Agreement 13/2016)		
UTFSM Credits: 3	Prerequisites: INF-253 INF-239	Exam: Does not have	Faculty.		
SCT Credits: 5			Computer Science Department		
Lecture Hours Weekly: 3	Weekly Assistantship Hours: 1.5	Weekly Laboratory Hours: 0	Semester in which it is taught		
			Odd	Pair X	Both
Formative axis: Applied Engineering - Software Development					
Total time dedicated to the subject: 150 chronological hours					

Subject Description

The student builds software using object-oriented analysis and design techniques. Develops a small 3-layer Web system, applying use case techniques for the analysis and specification of small project requirements, and object-oriented techniques (patterns and frameworks) for the design of software systems based on specifications. Practical activities strengthen teamwork.

Entry requirements

- Understand the object-oriented paradigm.
- Program in an object-oriented language (Java, C++ or others).
- Design data models and implement them in databases.
- Understand technical texts in English.

Contribution to the graduation profile

<p>Specific Competence</p> <ul style="list-style-type: none"> • Develop, implement and maintain reliable, efficient and feasible software systems. <p>Transversal Competencies</p> <ul style="list-style-type: none"> • Communicate oral and written information effectively within the organizations in which one works, as well as with entities in the environment. • Act with autonomy, flexibility, initiative, and critical thinking when facing professional problems. • Develop their work with solid criteria that allow you to ensure quality from a systemic perspective.
--

Learning Results expected to be achieved in this subject

- **Analyzes** various domains, **applying** object-oriented paradigms: inheritance and polymorphism.
- **Prepare** functional requirements specifications for Web systems, **applying them** in use cases.
- **Develop** structural and dynamic models of object-oriented software, using use case specifications.
- **Identifies** application opportunities for software design patterns, **applying them** to specific design situations.
- **Evaluates** software design models, **adjusting them** to requirements and intrinsic quality.
- **Describe** the logic used to arrive at the proposed solution, **arguing** its analysis and design decisions.

Thematic contents

<ul style="list-style-type: none"> • Object orientation: objects, types/polymorphism, classes/inheritance. • Software requirements: use cases, domain models. • Dynamic modeling: collaborations, sequences, states, activities. • Preliminary design: class models. • Design techniques: object-oriented frameworks, design patterns. • Design patterns for Web systems: presentation, concurrency, persistence. • Design quality: metrics, design reviews, traceability.

Teaching and learning methodology

<ul style="list-style-type: none"> • Expository classes supported by audiovisual media. • Learning based on the analysis and discussion of cases. • Learning based on the development of a project. • Independent study and student presentations on specific topics.

Evaluation and grading of the subject (Adjusted to Institutional Regulations-Regulation No. 1)

Approval and qualification requirements	<p>Evaluation and qualification process:</p> <p>The final grade is calculated as the weighted average of the grades of:</p> <ul style="list-style-type: none"> • two exams (C_1 and C_2) 30% each. • tasks associated with a project (T) 30% and, • participation and presentations (NP) - 10% <p>• Semiannual average (PS) is calculated according to: If $(C_1 + C_2)/2 \geq 50$ & $T \geq 50$ & $NP \geq 50$</p> $PS = C_1 \cdot 30\% + C_2 \cdot 30\% + T \cdot 30\% + NP \cdot 10\%$ <p>But</p> $PS = \min((C_1 + C_2)/2, T, NP)$
---	--

Resources for learning.

Virtual platform

Bibliography:

Guide Text	<ul style="list-style-type: none"> • Martin, Fowler. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley Professional. • Doug Rosenberg and Matt Stephens. (2007). Use Case Driven Object Modeling with UML: Theory and Practice. Press.
Complementary or Optional	<ul style="list-style-type: none"> • Alistair Cockburn. (2000). Writing Effective Use Cases Addison-Wesley Professional • Daryl Kulak and Eamonn Guiney. (2003). Use Cases: Requirements in Context (2nd Ed.) Addison-Wesley Professional. • Axel van Lamsweerde. (2009) Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley.

II. **CALCULATION OF NUMBER OF HOURS OF DEDICATION - (SCT-Chile) - SUBJECT SUMMARY TABLE**

ACTIVITY	Number of hours of dedication		
	Number of hours per week	Number of weeks	Total number of hours
PRESENCE			
Lecture or theoretical classes	3	fifteen	Four. Five
Assistantship/Exercises	1.5	14	twenty-one
Industrial visits (from Field)			
Laboratories / Workshop			
Evaluations (exams, others)	2	2	4
Others (specify)			
NO PRESENCE			
Assistantship			
Mandatory tasks	3	16	48
Individual and Group Work			
Others: Personal Study	2	16	32
TOTAL (HOURS)			150
Total number of TRANSFERABLE CREDITS			5