Universidad de Alcalá

TEACHING GUIDE

Software Patterns

Degree in
Information System Engineering (GISI)
Computer Engineering (GIC)
Computer Science Engineering (GII)

Universidad de Alcalá

Academic Year 2023/2024

4th Year - 1st Semester (GISI+GIC+GII)

Approved by the School Board on June 12th, 2023

TEACHING GUIDE

Course Name:	**Software Patterns**
Code:	**780042 (GISI+GIC+GII)**
Degree in:	**Information System Engineering (GISI)** **Computer Engineering (GIC)** **Computer Science Engineering (GII)**
Department and area:	**Ciencias de la Computación** **Computer Science**
Type:	**Optional (Generic) (GISI+GIC+GII)**
ECTS Credits:	**6.0**
Year and semester:	**4th Year - 1st Semester (GISI+GIC+GII)**
Teachers:	Por definir
Tutoring schedule:	Consultar al comienzo de la asignatura
Language:	English

1. COURSE SUMMARY

The subject software patterns aims teach students to use patterns in developing software applications as essential part of software engineering. The course focuses on the use of design patterns that represent a solution to problems that occur in a normal way in creating software.

Prerequisites and Recommendations

Is mandatory have studied subjects related to programming and software engineering.

2. SKILLS

Basic, Generic and Cross Curricular Skills.

This course contributes to acquire the following basic, generic and corss curricular skills:

en_CG4 - Ability to define, evaluate and select hardware and software platforms for the development and execution of computer systems, services and applications, in accordance with the knowledge acquired as set out in section 5, annex 2, of resolution BOE-A-2009 -12977.

en_CG5 - Ability to conceive, develop and maintain computer systems, services and applications using software engineering methods as an instrument for quality assurance, in accordance with the knowledge acquired as established in section 5, annex 2, of the resolution BOE-A-2009-12977.

en_CG8 - Knowledge of the basic subjects and technologies, which enable them to learn and develop new methods and technologies, as well as those that provide them with great versatility to adapt to new situations.

en_CG9 - Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to know how to communicate and transmit the knowledge, skills and abilities of the profession of Computer Engineering Engineer.

en_CG12 - Knowledge and application of basic elements of economics and human resources management, organization and planning of projects, as well as legislation, regulation and standardization in the field of computer projects, in accordance with the knowledge acquired as established in section 5 , annex 2, of resolution BOE-A-2009-12977.

en_CB1 - That students have demonstrated to possess and understand knowledge in an area of study that is based on general secondary education, and is usually found at a level that, although supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

en_CB2 - That the students know how to apply their knowledge to their work or vocation in a professional manner and possess the competencies that are usually demonstrated through the elaboration and defense of arguments and the resolution of problems within their area of study.

en_CB3 - That students have the ability to gather and interpret relevant data (usually within their area of study) to make judgments that include a reflection on relevant social, scientific or ethical issues.

en_CB4 - That students can transmit information, ideas, problems and solutions to both a specialized and non-specialized public.

en_CB5 - That the students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

en_TRU1 - Capacity of analysis and synthesis.

en_TRU2 - Oral and written competencies.

en_TRU3 - Ability to manage information.

en_TRU4 - Autonomous learning skills.

en_TRU5 - Team work.

Specific Skills

This course contributes to acquire the following specific skills:

en_CC1 - Ability to have a thorough knowledge of the fundamental principles and models of computing and know how to apply them to interpret, select, evaluate, model, and create new concepts, theories, uses and technological developments related to computer science.

en_CC3 - Ability to evaluate the computational complexity of a problem, know algorithmic strategies that can lead to its resolution and recommend, develop and implement the one that guarantees the best performance according to the established requirements.

en_CC4 - Ability to know the fundamentals, paradigms and techniques of intelligent systems and analyze, design and build systems, services and computer applications that use these techniques in any field of application.

en_CSI3 - Ability to actively participate in the specification, design, implementation and maintenance of information and communication systems.

Learning Outcomes

The expected learning outcomes, determined from the specific competences included in the verified memory of the degree as specific competences, are the following:

RA1. Define the different types of software patterns and its main features. Introduce the concept of design pattern and features.

RA2. Identify the fundamental patterns on which the design patterns are based and the GRASP patterns.

RA3. Describe the three families in which the design patterns are classified and identify the use of each of the patterns. Determine what is the most appropriate design pattern for a particular problem.

RA4. Know how to interpret the graphic model that represents the structure of a design pattern and be able to adapt it to the domain of a specific problem.

RA5. Analyze and design applications in which design patterns are applied.

RA6. Develop applications in object-oriented programming languages that include design patterns.

RA7. Knowing how to combine different design patterns to solve a complex software problem.

RA8. Define what is a framework and what is its application in the construction of software applications. Show the relationship that exists between design patterns and frameworks.

3. CONTENTS

Contents Blocks	Total number of hours
1. Introduction to software patterns: Origins, characteristics, classification, Patterns vs Frameworks.	4 hours
2. Fundamental patterns: Delegation, Interface, Abstract Superclass, Interface and Abstract Class, Immutable, Marker Interface, Proxy.	3 hours
3. GRASP patterns: Low coupling, High cohesion, Information expert, Creator, Controller, Polymorphism, Pure fabrication, Indirection, Protected variations	2 hours
4. Creational patterns: Introduction, Abstract Factory, Builder, Factory Method, Prototype, Singleton.	10 hours
5. Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy.	12 hours
6. Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor.	18 hours
7. Antipatterns: Software Development, Software Architecture, Software Project Management.	3 hours
8. Frameworks: Frameworks features, frameworks types, examples of frameworks.	4 hours

4. TEACHING - LEARNING METHODOLOGIES. FORMATIVE ACTIVITIES.

4.1. Credits Distribution

Number of on-site hours:	58 hours (56 hours on-site +2 exams hours)
Number of hours of student work:	92
Total hours	150

4.2. Methodological strategies, teaching materials and resources

The subject Software Patterns is organized as a four-month course of 6 ECTS (150 hours).

In the teaching-learning process of the aforementioned contents, the following training activities will be used:

- Classroom theoretical classes.
- Practical classes: solving classroom problems.

- Classroom laboratory practices.
- Tutorials: individual and / or group.

In addition, depending on the nature of the different parts of the subject matter, the following training activities may be used, among others:

- Preparation of works with individual responsibility but with information management as a team.
- Put in common the information, problems and doubts that appear in the realization of the works.
- Organization and realization of public days with oral presentations and discussion of results.
- Use of eLearning platform (Aula Virtual).

Classroom activities:

1. In the classroom: exhibition and discussion of the basic knowledge of the subject. Approach and theoretical resolution of exercises and related assumptions. Oriented to the teaching of the specific competences of the subject, especially those related to knowledge and techniques of using software patterns in the creation of computer applications.
2. In the laboratory: planning and development of practical exercises to solve problems and analyze hypotheses and contribute to the development of the ability to analyze results, critical reasoning and understanding of the proposed resolution methods. They will serve as the basis for the acquisition of the generic competences described in section 2.

Not in-person activities:

1. Analysis and assimilation of the contents of the subject, resolution of problems, bibliographical consultation, preparation of individual and group work, realization of face-to-face exams and self-evaluations. Specially oriented to the development of methods for self-organization and planning of individual and team work.
2. Tutorials: individual and group counseling during the teaching-learning process, either in person or remotely.

5. ASSESSMENT: procedures, evaluation and grading criteria

Preferably, students will be offered a continuous assessment model that has characteristics of formative assessment in a way that serves as feedback in the teaching-learning process.

5.1. PROCEDURES

The evaluation must be inspired by the criteria of continuous evaluation (Learning Assesment Guidelines, LAG, art 3). However, in compliance with the regulations of the University of Alcalá, an alternative process of final evaluation is made available to the student in accordance with the Learning Assesment Guidelines as indicated in Article 10, students will have a period of fifteen days from the start of the course to request in writing to the Director of the Polytechnic School their intention to take the non-continuous evaluation model adducing the reasons that they deem convenient. The evaluation of the learning process of all students who do not apply for it or are denied it will be done, by default, according to the continuous assessment model. The student has two calls to pass the subject, one ordinary and one extraordinary.

In the ordinary call the default evaluation method is the "continuous evaluation", with characteristics of formative evaluation to serve as feedback in the teaching-learning process by the student.

Optionally, and in a justified manner, the student may request the evaluation from the director of the center, by means of a single exam, which must be requested in writing and within the prescribed time limits. This method of evaluation consists of a theoretical examination and a laboratory practice.

In the extraordinary period, the evaluation will be based on a single exam, consisting of a theoretical exam and a laboratory practice, in which the degree of mastery of the competences of the subject will be determined.

Evaluation Criteria

The performance of the students will be evaluated according to the knowledge and skills acquired. The methods to be used will be: the resolution of practical cases and proposed works, public defense of certain works, realization of a final practice, as well as the performance of exams to check the theoretical knowledge of the subject.

Each of the continuous assessment exams will consist of two parts Theory and Laboratory. Distribution of the qualification in the Continuous Evaluation Exams (PEC's):

- The theoretical part will be valued at 50% and that of laboratory practices at another 50%, to obtain 100% of the grade of the subject.
- For the theoretical part, two PEI (Intermediate Assessment Test) will be carried out, having a weight of 10% and 40% (PEI1 - PEI2) respectively.
- For the laboratory practices part, two PL (Laboratory Assessment Test) will be carried out, having a weight of 10% and 40% (PL1 - PL2) respectively.

In the Extraordinary Evaluation, the evaluation will be based on two exams, consisting of a theoretical exam and a laboratory practice (with an assessment of 50% each), in which the degree of mastery of the competences of the subject will be determined.

Students who take part in Final Evaluation must complete two exams, consisting of a theoretical exam and a laboratory practice (with an assessment of 50% each), in which the degree of mastery of the competences of the subject will be determined.

5.2. EVALUATION

EVALUATION CRITERIA

The following specific evaluation criteria are established for the subject:
- CE1. The student knows the different types of software patterns and their characteristics as well as the concept of design pattern.
- CE2. The student knows how to identify the fundamental patterns and the GRASP patterns.
- CE3. The student knows the three families of design patterns and understands the applicability of each of the patterns.
- CE4. The student has acquired the necessary knowledge to adapt the general structure of a design pattern to a specific problem.
- CE5. The student has acquired the necessary knowledge to apply design patterns in the development of a computer application.
- CE6. The student knows how to apply the design patterns in an object-oriented language.
- CE7. The student understands what is a framework and its relation with the design patterns.

GRADING CRITERIA

In the following table the weight in the qualification (between 0 and 100) of each exam is indicated, and its relation with the evaluation criteria, learning results and general competences.

In the ordinary call-continuous assessment the relationship between the competences, learning outcomes, criteria and evaluation instruments is as follows.

Skill	Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
CG4, CG5, CG8, CG9, CG12. CC1, CC4, CIC4, CSI3	RA1, RA2, RA3, RA4, RA5, RA8	CE1, CE2, CE3, CE4, CE5	PEI1, PEI2	50%
	RA4, RA5, RA6, RA7	CE2, CE3, CE6	PL1, PL2	50%

For the final and extraordinary evaluation will have to perform two exams, one with the theoretical content (PT) and another with the laboratory (PL) having a weight of 50% each.

Skill	Learning Outcomes	Evaluation criteria	Grading Tool	Contribution to the final mark
CG4, CG5, CG8, CG9, CG12. CC1, CC4, CIC4, CSI3	RA1, RA2, RA3, RA4,RA5, RA8	CE1, CE2, CE3, CE4, CE5	PT	50%
	RA1, RA2, RA3, RA4, RA5, RA6	CE2, CE3, CE6	PL	50%

6. BIBLIOGRAPHY

6.1. Basic Bibliography

Patrones de Diseño
Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
Addison Wesley, 2003

Patrones de diseño aplicados a Java
Stephen Steling, Olav Maassen
Pearson Education / Sun Microsystems, 2003

Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software
Eric Freeman, Elisabeth Freeman
O´Reilly, 2020

Software Architecture Design Patterns in Java
Partha Kuchana
Auerbach, 2004

Patterns in Java, volumen 1 y 2
Mark Grand
Wiley computer publishing, 1998

Java Design Patterns: A Tutorial
James W. Cooper
Addison-Wesley, 2000

Patrones de diseño en Java
Laurent Debrauwer
Ediciones Eni, 2013

Introducción a los patrones de Diseño
Oscar Blancarte, 2016
https://www.oscarblancarteblog.com/libros/introduccion-los-patrones-diseno/

6.2. Additional Bibliography

Construcción de Software Orientado a Objetos. Segunda Edición.
Bertrand Meyer. Prentice Hall, 1998.

Core J2EE™ Patterns: Best Practices and Design Strategies
Deepak Alur, John Crupi, Dan Malks. Prentice Hall / Sun Microsystems Press, 2001

Disclosure Note

During the evaluation tests, the guidelines set out in the Regulations establishing the Rules of Coexistence of the University of Alcalá must be followed, as well as the possible implications of the irregularities committed during said tests, including the consequences for committing academic fraud according to the Regulation of Disciplinary Regime of the Students of the University of Alcalá.