



ACADEMIC PROGRAM

SOFTWARE DESIGN

B.F.A. IN COMPUTER SCIENCE

MODALITY: ON CAMPUS

ACADEMIC YEAR: 2022-2023

Name of the course:	Software Design
Degree :	Computer Science
Location:	Centro Universitario de Tecnología y Arte Digital
Area:	Software Engineering
Year:	2º
Teaching period:	2
Type:	OB
ECTS credits:	6
Teaching modality:	On campus
Language:	English
Lecturer / Email	-
Web page:	http://www.u-tad.com/

SUBJECT DESCRIPTION

Area description

This subject establishes the knowledge and techniques necessary for the correct specification, design and implementation of software projects based on good engineering practices and methodologies.

Subject description

Model and design solutions based on commitments to efficiency, modularity, quality and maintainability. Understand and design software architectures based on object orientation, using design techniques and patterns. It is a fundamental subject in order to understand how to create better software

COMPETENCIES AND LEARNING OUTCOMES

Competencies

BASIC AND GENERAL COMPETENCIES

CB1: That students have demonstrated possession and understanding of knowledge in an area of study that builds on the foundation of general secondary education, and is usually at a level that, while relying on advanced textbooks, also includes some aspects that involve knowledge from the cutting edge of their field of study.

CB2: That 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 problem solving within their area of study.

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.

CB4: Students should be able to transmit information, ideas, problems and solutions to both specialized and non-specialized audiences.

CB5: That students have developed those learning skills necessary to undertake further studies with a high degree of autonomy.

CG1 - Ability to understand, plan and solve problems through the development of IT solutions.

GC2 - Development of IT solutions respectful of the environment, social duties and natural resources, in addition to complying with legislation and ethics.

CG3 - Knowledge of the scientific fundamentals applicable to the resolution of computer science problems.

CG6 - Integration, as a software engineer, in multidisciplinary work environments demonstrating teamwork skills, versatility, flexibility, creativity and respect for the work of colleagues from other areas.

CG9 - Ability to learn, modify and produce new computer technologies.

CG10 - Use of creative techniques for the realization of computer projects.

SPECIFIC COMPETENCES

CE10 - Ability to manage a code versioning manager and generate the documentation side of an application automatically.

SC18 - Ability to design the architecture of an object-oriented computer application using the most appropriate design patterns and integrating them into the complete architecture.

SC19 - Ability to conceive, design through graphic languages and implement a computer application using different development methodologies, from the conception of the product to its final development through the definition of its phases and iterations.

CE20 - Ability to test the operation and functionality of a computer application, elaborating test plans and using test-oriented design and programming techniques.

CE22 - Knowledge of the techniques and implications of computer application maintenance including those that use reverse engineering principles to understand and modify a software whose structure is unknown.

Learning outcomes

Upon completion of the degree, the graduate will be able to:

- To know formal specification languages
- To be able to identify and apply design patterns in problem solving
- To use refactoring techniques
- To understand the software life cycle

- To understand and apply waterfall methodologies in software development
- To understand and apply Scrum in software development

CONTENTS

Software specification languages

Design patterns

Refactoring techniques

SUBJECT SYLLABUS

Topic 1_ Software description languages

Topic 2_ Design patterns

- STRATEGY pattern
- OBSERVER pattern
- DECORATOR pattern
- STATE pattern
- FACTORY METHOD pattern
- ABSTRACT FACTORY pattern
- SINGLETON pattern
- ADAPTER pattern
- FACADE pattern
- TEMPLATE METHOD pattern

Topic 3_ Refactoring techniques

TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

TRAINING ACTIVITIES

LEARNING ACTIVITIES	Total hours	Hours of presence
<i>Theoretical / Expository classes</i>	25,00	25,00
<i>Practical classes</i>	29,00	29,00
<i>Tutorials</i>	4,00	2,00
<i>Independent study and autonomous work of the student</i>	50,00	0,00

<i>Elaboration of work (group or individual)</i>	32,00	0,00
<i>Evaluation Activities</i>	10,00	10,00
TOTAL	150	66

Teaching methodologies

Expository method or master lesson

Case learning

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

Just in time Teaching (JITT) or classroom on time

Expository method or master lesson

Case method

Learning based on problem solving

Cooperative or collaborative learning

inquiry learning

Flipped classroom methodology

Gamification

TEMPORAL DEVELOPMENT

DIDACTIC UNITS / TOPICS TIME PERIOD

Topic 1_Software description languages Week 1

Topic 2_Design patterns: Introduction Week 2

Topic 2_STRATEGY Pattern Week 2, 3

Topic 2_OBSERVER Pattern Week 4, 5

Topic 2_DECORATOR Pattern Week 6

Topic 2_STATE Pattern Week 7
 Topic 2_FACTORY METHOD Pattern Week 8
 Topic 2_ABSTRACT FACTORY Pattern Week 9
 Topic 2_SINGLETON Pattern Week 10
 Topic 2_ADAPTER Pattern Week 11
 Topic 2_FACADE Pattern Week 12
 Topic 2_TEMPLATE METHOD Pattern Week 13
 Topic 3_Refactoring techniques Weeks 14 and 15

EVALUATION SYSTEM

ASSESSMENT SYSTEM	MINIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)	MAXIMUM SCORE RESPECT TO THE FINAL ASSESSMENT (%)
<i>Assessment of participation in class, exercises or projects of the course</i>	0	30
<i>Assessment of assignments, projects, reports, memos</i>	30	80
<i>Objective test</i>	10	60

GRADING CRITERIA

ASSESSMENT SYSTEM	ORDINARY EVALUATION	EXTRAORDINARY EVALUATION
<i>Assessment of participation in class, exercises or projects of the course</i>	10	0
<i>Assessment of assignments, projects, reports, memos</i>	30	30
<i>Objective test</i>	60	70

General comments on the evaluations/assessments

The evaluation will be done through a final exam in the ordinary call where it will be essential to obtain a minimum grade of 4 in order to be able to average it with the project and continuous evaluation grades. The exam will be taken without a computer.

. Regarding the extraordinary call, the exam will count for 70% of the grade and the project for 30%, with it being again mandatory to obtain more than a 4 in the exam to make the average. The project grade will be saved if it has been carried out in the ordinary call.

. Work out of form and date will not be accepted without justified cause, each delivery is understood as an exam and will have the right to review. And if they are accepted it will be with a considerable reduction in the grade.

LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Basic Bibliography:

- Freeman, E.; Freeman, E.; Bates, B. and Sierra, K. (2004); HEAD FIRST DESIGN PATTERNS; Editorial. O'Reilly; ISBN: 0596007124
- Debrauwer, L. (2013); DESIGN PATTERNS IN JAVA: THE 23 DESIGN PATTERNS: DESCRIPTION AND SOLUTION ILLUSTRATED IN UML 2 AND JAVA; Publisher: ENI; ISBN: 9782746086456
- Fowler, M. (1999); REFACTORING: IMPROVING THE DESIGN OF EXISTING CODE; Publisher: ADDISON-WESLEY; ISBN: 9780201485677

Recommended Bibliography:

- Gamma, E.; Johnson, R.; Helm, R. and Vlissides, J. (1994); DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE; Addison-Wesley; ISBN: 0-201-63361-2
- Shalloway, Alan; Trott, James; Design Patterns Explained: A New Perspective on Object-Oriented. Addison-Wesley Professional (2001). ISBN 10: 0201715945 ISBN 13: 9780201715941
- Steven John Metsker; The design patterns Java workbook; Addison Wesley, 2002. ISBN: 0-201-74397-3
- James W. Cooper; Java™ Design Patterns: A Tutorial; Addison Wesley, 2000. ISBN: 0-201-48539-7

REQUIRED MATERIALS, SOFTWARE AND TOOLS

Type of classroom

Theory classroom

Board and projection system

Materials:

Computer with Windows

Software:

Eclipse