ACADEMIC PROGRAM

PARADIGMS OF PROGRAMMING

B.F.A. IN **COMPUTER SCIENCE**

MODALITY: ON CAMPUS

ACADEMIC YEAR: 2022-2023

Name of the course:	Paradigms of Programming
Degree :	Computer Science
Location:	Centro Universitario de Tecnología y Arte Digital
Area:	Programming
Year:	4º
Teaching period:	1
Type:	OB
ECTS credits:	3
Teaching modality:	On campus
Language:	English
Lecturer / Email	-
Web page:	http://www.u-tad.com/

SUBJECT DESCRIPTION

Area description

This subject belongs to the programming subject. This subject is dedicated to the study of programming techniques and languages on which the software engineering degree studies will be based.

Subject description

This subject allows you to understand the different programming paradigms that currently exist, showing the alternatives available when making decisions for software development. Throughout the course, the main programming paradigms of the last decades have been seen: Structured programming, procedural programming, object-oriented programming, concurrent programming... In this subject we will focus on two new approaches: Functional programming and Reactive programming

The focus of the course will therefore be mainly practical, based on the exhaustive development of examples on the most common development environment at the moment, which is Python and its specialized libraries.

COMPETENCIES AND LEARNING OUTCOMES

Competencies

BASIC AND GENERAL SKILLSs

CG1 - Ability to understand, schedule and solve problems trough software development

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

CG4 - Ability to simplify and optimize computer systems by understanding their complexity

CG9 - Ability to learn, modify and develop new software solutions

CG10 - Use of creative techniques to carry out computer projects

CB1 That students have demonstrated knowledge and understanding in an area of study that starts from the basis of general secondary education, and is usually at a level that, although it is supported by advanced textbooks, also includes some aspects that involve knowledge from the forefront of their field of study.

CB2 Students are able to apply their knowledge to their work or vocation in a professional manner and possess the competences usually demonstrated through the development and defense of arguments and problem solving within their field of study.

CB3 That students have the ability to gather and interpret relevant data (usually within their area of study) in order to make judgements that include reflection on relevant social, scientific or ethical issues.

CB4 Students are able to convey information, ideas, problems and solutions to both specialist and non-specialist audiences.

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

SPECIFIC SKILLS

CE1 - Knowledge of the structure of computers, the concepts of coding, manipulation, information processing and use of low-level languages

CE7 - Knowledge of the main types of data structures and use of libraries and algorithmic techniques associated with these structures together with the complexity orders that characterize these techniques

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación/ Knowledge of the different software

paradigms that underpin programming languages

CE9 - Knowledge of effective control structures, variables, programming syntax and memory management in the development of a computer application

CE10 - Ability to work with a release manager and generate application documentation automatically.

CE15 - Knowledge of fault tolerance, adaptability, load balancing and system predictability for distributed application development

CE17 - Knowledge of the parallelization characteristics of graphics cards and high-performance architectures for application development.

CE20 - Ability to test the operation and functionality of a computer application, develop test plans and use test-oriented design and programming techniques

CE23 - Knowledge of the principles of artificial intelligence and use of deterministic search algorithms and state machines

Learning outcomes

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

- To understand and handle the concept of dynamic memory
- To identify classes with the relevant data of a problem
- To instance classes and objects and manage them
- To understand and use the mechanisms of inheritance, polymorphism and operator overloading
- To identify class relationships in different use-cases.
- To master an object oriented programming language.
- To master programming patterns
- To know different problem solution strategies from an algorithmic view point: divide and conquer, dynamic programming, backtracking or genetic algorithms.
- To understand algorithmic complexity, assess it and search for optimal solutions
- To code a program able to find the optimal path between any pair of nodes of a graph
- To build neural networks to solve applied problems

CONTENTS

Programming paradigms

SUBJECT SYLLABUS

1. Introduction to programming paradigms

1.1. Introduction to Python

2. Functional Programming

2.1. Practice 1

3. Reactive Programming

3.1. Practice 2

TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

TRAINING ACTIVITIES

LEARNING ACTIVITIES	Total hours	Hours of presence
Theoretical / Expository classes	17,82	17,82
Practical classes	9,45	9,45

Tutorials	2,00	1,00
Independent study and autonomous work of the student	25,91	0,00
Elaboration of work (group or individual)	16,91	0,00
Evaluation Activities	2,91	2,91
TOTAL	75	31,18

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

1. Introduction to programming paradigms Week 1 and 2

2. Functional Programming Weeks 3, 4, 5, and 6

3. Reactive Programming Weeks 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15

EVALUATION SYSTEM

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

GRADING CRITERIA

ASSESSMENT SYSTEM	ORDINARY EVALUATION	EXTRAORDINARY EVALUATION
Assessment of participation in class, exercises or projects of the course	0	0
Assessment of assignments, projects, reports, memos	80	80
Objective test	20	20

General comments on the evaluations/assessments

● The practices will be completely individual, and will be accompanied by an explanatory report. In the event that there are doubts regarding the qualification, an oral defense of the same will be proposed.

● In order to successfully pass the subject, it will be necessary for the student to obtain a 4 in each practice and in the final exam. An average will not be taken if all requirements are not met. The total average must exceed 5.

● If the student does not obtain approval in the ordinary call, the student must re-sit the failed parts, maintaining the same evaluation criteria for the extraordinary call.

LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Programming Languages: Principles and Paradigms

Maurizio Gabbrielli , Simone Martini (2010)

REQUIRED MATERIALS, SOFTWARE AND TOOLS

Type of classroom

Theory classroom

Board and projection system

Materials:

Personal Computer

Software:

-