ACADEMIC PROGRAM

ANALYSIS AND DESIGN OF ALGORITHMS

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

MODALITY: ON CAMPUS

ACADEMIC YEAR: 2022-2023

Name of the course:	Analysis and Design of Algorithms
Degree :	Computer Science
Location:	Centro Universitario de Tecnología y Arte Digital
Area:	Programming
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 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

In this subject, the analysis and design of algorithms is studied from a theoretical and practical point of view.

The analysis of the temporal and spatial complexity of an algorithm and the implications of these results in its execution will be delved into.

Different algorithms will be studied for solving various problems, emphasizing, at all times, the advantages and disadvantages of each of them, as well as the contexts in which it should be applied.

Additionally, case studies will be carried out in which the algorithms will be programmed and their effectiveness in solving problems will be verified.

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

Graph algorithms

Ordering algorithms

Search algorithms

Amortizaed analysis

Algorithmic patterns: Backtrackind, Divide and Conquer, Dymanic Programming

SUBJECT SYLLABUS

• Topic 0. Presentation

Algorithmics

• Topic 1. Introduction to software construction

o Preconditions, Nomenclature, Comments

o Good Programming Principles

• Topic 2. Non-recursive algorithms

o Temporal and Spatial Complexity in Iterative Algorithms

o Sorting Algorithms: Bubble, Insertion and Selection

• Topic 3. Recursive algorithms

o Recursion

o Temporal and Spatial Complexity in Recursive Algorithms

• Topic 4. Programming Strategies

o Dynamic Programming

o Divide and Conquer

or Backtracking

o Greedy Algorithm

Data structures.

• Topic 5. Lists

o Linked and Contiguous List

o Stacks and Queues

• Topic 6. Hash Tables

o Hashing, Implementation, Collision Resolution

• Topic 7. Trees

o Binary Search Trees, AVL

or Mounds

• Additional Topics

or Graphs

o Genetic Algorithms

TRAINING ACTIVITIES AND TEACHING METHODOLOGIES

TRAINING ACTIVITIES

LEARNING ACTIVITIES	Total hours	Hours of presence
Theoretical / Expository classes	35,64	35,64
Practical classes	18,91	18,91
Tutorials	4,00	2,00
Independent study and autonomous work of the student	51,82	0,00
Elaboration of work (group or individual)	33,82	0,00
Evaluation Activities	5,82	5,82

TOTAL	150	62,37

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

Introduction / Review C++ 1 week

Classes / Non-recursive algorithms 1 week

Ordination 1 week

Recursion 1 week

Algorithmic schemes: backtracking, divide and conquer, dynamic programming 2 weeks

Contiguous/Linked Lists 1 week

Circular Lists / Stacks / Queues 1 week

Hash tables 1 weeks

Trees 2 weeks

Additional Topics 1 week

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	10	0
Assessment of assignments, projects, reports, memos	60	70
Objective test	30	30

General comments on the evaluations/assessments

• Final numerical grade will be from 0 to 10, with 5 being the minimum grade to pass.

• It will be necessary to pass both the exam and the practices separately with a minimum grade of 5

• Global evaluation of the learning process and acquisition of skills and knowledge.

• Active participation will be required from the student, necessary for the development of the classes.

• Those students who fail any practice or who submit them after the designated time may repeat them until they pass, although they will receive a penalty in their grade. The deadline for submission will be the last day of class.

LIST OF REFERENCES (BOOKS, PUBLICATIONS, WEBSITES):

Basic:

Hernández, R., Lázaro, J.C., Dormido, R. and Ros, S. (2001). Data structures and algorithms. Ed. Prentice Hall. ISBN: 978-84-205-2980-6

Martí, N., Ortega, Y. and Verdejo, J.A. (2003). Data structures and algorithmic methods: solved exercises. Ed. Pearson Alhambra.

Rodríguez, M. and Gonzalo, J. (1998). Algorithmic schemes: methodological approach and solved problems. UNED Ed. ISBN: 9788436236224.

Hernández, R., Carmona, E.J., Martínez, R. and Pastor, R. (2006). Problems of data structures and algorithms. Ed. Ramón Areces. ISBN: 84-8004-723-2.

Murphy, J. (2005). The visual investor. Ed. Netbiblo. ISBN: 978-8497451017

Recommended:

Guerrequeta, R. and Vallecillo, A. (1998). Algorithm design techniques. Ed. University of Malaga.

Brassard, G., & Bratley, P. (1997). Algorithm Fundamentals. Ed. Prentice Hall. ISBN: 0-13-335068-1.

Edwards, R. D. and Magee J. (2010). Technical analysis of stock trends. Ed. BN Publishing. ISBN: 978-1607960799

Serafini, M.T. (2007). How is this written. Ed. Paidós. ISBN: 978-84-493-1959-4. It is to learn how to write, in case any student needs it.

REQUIRED MATERIALS, SOFTWARE AND TOOLS

Type of classroom
Theory classroom

Board and projection system

Materials:
Computer with Windows, macOS or Linux

Software:
Para acostumbrarse al uso de las herramientas que se dispondrán en elexamen, se aconseja el siguiente entorno:

Terminal posix: Cygwin

Compilador gcc/g++, ejecutado por terminal

Editor Notepad++

En general, cualquier entorno de programación que soporte C