DEGREE CURRICULUM

COMPUTING TECHNIQUES

Coordination: OJEDA CONTRERAS, JESUS

Academic year 2022-23

COMPUTING TECHNIQUES 2022-23

Subject's general information

<table>
<tr><td>Subject name</td><td>COMPUTING TECHNIQUES</td></tr>
<tr><td>Code</td><td>102376</td></tr>
<tr><td>Semester</td><td>1st Q(SEMESTER) CONTINUED EVALUATION</td></tr>
<tr><td>Typology</td><td>
<table>
<tr><th>Degree</th><th>Course</th><th>Character</th><th>Modality</th></tr>
<tr><td>Bachelor's degree in Digital Interaction and Computing Techniques</td><td>2</td><td>COMPULSORY</td><td>Attendance-based</td></tr>
</table>
</td></tr>
<tr><td>Course number of credits (ECTS)</td><td>6</td></tr>
<tr><td>Type of activity, credits, and groups</td><td>
<table>
<tr><th>Activity type</th><td>PRALAB</td><td>TEORIA</td></tr>
<tr><th>Number of credits</th><td>3</td><td>3</td></tr>
<tr><th>Number of groups</th><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Coordination</td><td>OJEDA CONTRERAS, JESUS</td></tr>
<tr><td>Department</td><td>COMPUTER SCIENCE AND INDUSTRIAL ENGINEERING</td></tr>
<tr><td>Teaching load distribution between lectures and independent student work</td><td>6 ECTS = 25x6 = 150 hours of work
40% -> 60 in-class hours
60% -> 90 autonomous work hours</td></tr>
<tr><td>Important information on data processing</td><td>Consult this link for more information.</td></tr>
<tr><td>Language</td><td>Spanish, Catalan</td></tr>
</table>

COMPUTING TECHNIQUES 2022-23

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
OJEDA CONTRERAS, JESUS	jesus.ojedacontreras@udl.cat	6	

Subject's extra information

For any doubt and/or question, you can send an email to the teacher of the subject.

Learning objectives

- Be able to formally characterize problems.
- Analyze the efficiency of algorithms by using asymptotic notation.
- Identify the typology of the problem, as well as the most suitable algorithmic strategy for its resolution.
- Design and implement adequate data structures to represent the information of each problem.
- Design and implement efficient algorithmic strategies to solve the different types of problems.

Competences

- CT3. Implement new technologies and information and communication technologies.
- CG10. Ability to apply appropriate algorithmic techniques for solving computational problems.
- CE2. Ability to understand and master the basic concepts of discrete mathematics, logic, algorithmics and computational complexity, and its application for solving problems typical of computing.
- CE8. Knowledge and application of the basic algorithmic procedures of computer technologies to design solutions to computational problems, analyzing the suitability and complexity of the proposed algorithms.
- CE9. Know, design and efficiently use the most appropriate data types and structures to solve a problem.
- CE10. Ability to analyze, design, build and maintain interactive digital applications in a robust, safe and efficient way, choosing the most appropriate paradigm and programming languages.

Subject contents

1. Preliminary: Notation and definitions
2. Formal specification of algorithms
    - Pre-conditions, Post-conditions, Invariants
    - Algorithmic efficiency, asymptotic notation
    - Iteration and recursion
3. Algorithmic Paradigms
    - Greedy algorithms
    - Divide and conquer
    - Backtracking
    - Dynamic programming
4. Computational Complexity: P, NP, ...

Methodology

According to the schedule of the subject, each week the student attends 2 hours of Theory and 2 hours of laboratory (PRALAB).

The Theory sessions present the topics that can be consulted in the content section. They incorporate illustrative

examples and problem proposals to solve in the laboratory classes.

PRALAB sessions are taught in the laboratory and present problems and discuss the proposed solutions. They can also present the practices of the subject and carry out the corresponding laboratory work.

The student's autonomous work consists of solving the proposed exercises and the practical tasks when indicated.

The programming language used in class is Python 3.9+.

Development plan

Shem	Description	Theory Activity	PRALAB Activity	Autonomous work
1	Preliminaries	T1: Preliminaries	Review Python	Bibliography and program consultation, Python Review
2	Specification	T2: Specification	Problems T2	T2 problems
3	Specification	T2: Specification	Problems T2, Presentation P1	P1, T2 problems
4	Algorithmic efficiency	T2: Specification	P1	P1, T2 problems
5	Iteration	T2: Specification	P1	P1, T2 problems
6	Recursion	T2: Specification	T2 problems	P1, T2 problems
7	Recursion	T2: Specification	P1	P1, T2 problems
8	Paradigms: Intro	T3: Paradigms	Delivery P1, Doubts T2	Study
9		1st Partial		Study
10	Greedy algorithms	T3: Paradigms	Presentation P2	P2, T3 problems
11	Divide and conquer	T3: Paradigms	T3 problems	P2, T3 problems
12	Backtracking	T3: Paradigms	P2	P2, T3 problems
13	Dynamic programming	T3: Paradigms	P2	P2, T3 problems
14	Complexity	T4: Complexity	Problems T3, T4	P2, Problems T4
15	Complexity	T4: Complexity	Delivery P2, Doubts T3 and T4	T3 problems
16/17		2nd Partial		Study
18				
19		Recovery		Study

Evaluation

Acr	Assessment activity	Weight	Minimum Grade	In group	Mandatory	Recoverable
PE1	1st Partial Exam	25%	-	No	No	Yes
PE2	2nd Partial Exam	25%	-	No	No	Yes
P1	Practice 1	25%	-	Yes (<= 2)	No	No
P2	Practice 2	25%	-	Yes (<= 2)	No	No

Final Grade = 0.25 *PE1 + 0.25* PE2 + 0.25 *P1 + 0.25* P2

Recovery of written tests 1 and 2: If the final grade obtained in the course is <5, then the student can choose to improve / recover the 50% that the written tests represent (the student will be able to choose which part they want to recover, or to choose both parts).

Except for a new exceptional situation, the written tests will be face-to-face.

Bibliography

- Gilles Brassard, Paul Bratley. Fundamentals of Algorithmics. Prentice Hall. 1996.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (3rd ed). MIT Press. 2009.
- Steven S. Skiena. The Algorithm Design Manual. Springer. 2008.
- Cristopher Moore, Stephan Mertens. The Nature of Computation. Oxford University Press. 2011.