DEGREE CURRICULUM

ALGORITHMS AND PROGRAMMING

Coordination: MARTINEZ RODRIGUEZ, SANTIAGO

Academic year 2023-24

ALGORITHMS AND PROGRAMMING 2023-24

Subject's general information

Subject name	ALGORITHMS AND PROGRAMMING
Code	102364
Semester	1st Q(SEMESTER) CONTINUED EVALUATION
Typology	<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>1</td><td>COMMON/CORE</td><td>Attendance-based</td></tr></table>
Course number of credits (ECTS)	6
Type of activity, credits, and groups	Only examination
Coordination	MARTINEZ RODRIGUEZ, SANTIAGO
Department	COMPUTER ENGINEERING AND DIGITAL DESIGN
Teaching load distribution between lectures and independent student work	6 ECTS = 25x6 = 150 working hours. There are no classes for this subject.
Important information on data processing	Consult this link for more information.
Language	Catalan.
Distribution of credits	There are no classes for this subject.

ALGORITHMS AND PROGRAMMING 2023-24

Teaching staff	E-mail addresses	Credits taught by teacher	Office and hour of attention
MARTINEZ RODRIGUEZ, SANTIAGO	santi.martinez@udl.cat	0	Arrange with the teacher. Optionally, by videoconference.

Subject's extra information

To address the subject is advisable to show interest in analyzing real problems and developing technological solutions to solve them. It is also advisable to show analytical skills, logical reasoning and critical capacity.

The knowledge and competencies acquired in this subject will be useful to follow other subjects with contents related with programming languages, data structure and algorithms.

Learning objectives

The learning objectives of the course are to design algorithms, and then to implement these algorithms with a programming language. Specifically, the programming language used for this purpose is ANSI C/C++ and the problems to be solved are mainly those related with sequences processing.

In particular, the main learning objectives are:

- To design and implement algorithmic structures to solve the different types of problems.
- To design and implement data structures to encode information.
- To design and implement iterative algorithms.
- To identify problem types and to apply appropriate algorithmic strategies.
- To design and implement algorithms to solve complex problems in a structured way.
- To design and implement solutions using the top-down design technique.
- To use a software development environment based on a high-level programming language.

Competences

Basic Competences

- **B01**. That students have demonstrated to possess and understand knowledge in an area of study that starts from the base of general secondary education, and is usually found at a level that, although supported by advanced textbooks, also includes some aspects that imply knowledge coming from the vanguard of his/her field of study.

Transversal Competences

- **CT3**. Acquire training in the use of new technologies and information and communication technologies.
- **CT5**. Acquire essential notions of scientific thought.

General Competences

- **CG2**. Design, develop, evaluate and guarantee the accessibility, ergonomics, usability and security of computer systems.
- **CG3**. Use adequate hardware and software platforms to develop and execute interactive digital applications.
- **CG5**. Know the basic subject areas and technologies needed to learn and develop new methods and technologies, and those that help to adapt to new situations.
- **CG7**. Solve problems through initiative, determination, independence and creativity.
- **CG8**. Capacity for abstraction and critical, logical and mathematical reasoning.

Specific Competences

- **CE2**. Capacity to understand and master the basic concepts of discrete mathematics, logics, algorithmic and computational complexity, and its application to solve computational problems.
- **CE3**. Basic knowledge of the use and programming of computers, operating systems and databases, and their use in the development of interactive applications.
- **CE4**. Capacity to know, understand and evaluate the structure and architecture of computers, as well as the basic components that conform them.
- **CE16**. Capacity to design and evaluate person-computer interfaces that guarantee the usability of systems, services and

computer applications.
- **CE17**. Capacity to apply knowledge on design to propose and defend a design concept for an interactive system and use proper creative technologies to develop each project.
- **CE24**. Capacity to understand the human factors involved in any interactive process between humans and technology, as well as being able to adequately apply them in the design of interactive products and services, and their interfaces.

Subject contents

Introduction: Processes, algorithms and programs.

Unit 1. Basic algorithmic structures

1.1 Constants, variables, basic types and valid expressions
1.2 Assignment, sequential composition, alternative composition and iterative composition
1.3 Programming environment

Unit 2. Design of iterative programs

2.1 Sequential access

- Algorithmic schemes for sequence processing
- Algorithmic schemes for searching in sequences

2.2 Direct access. Tables

- Sequential tables
- Direct tables
- Multidimensional tables
- Classic sorting algorithms

Unit 3. Non-basic data processing

3.1 Procedures and functions
3.2 Parameter passing mechanisms
3.3 Top-down design of algorithms

Methodology

There are no classes for this subject.
However, the student who wishes may attend, as an auditor, the equivalent subject (Introduction to Programming I) of the Degree in Computer Engineering that is taught on the same Igualada campus.

Development plan

There are no classes for this subject.

Evaluation

The **continuous evaluation** of the subject is based on 3 blocks:

- **Practice Block** (25%): It consists of two activities: Practice 1 and Practice 2. They cannot be improved. A minimum grade is not required.
- **Theory block 1** (25%): It consists of one activity: 1st Midterm Exam. It can be improved with the theory block 2. A minimum grade is not required. Date of the exam: the date of the realization of the 1st Midterm Exam, defined by the EPS.
- **Theory block 2** (50%): It consists of one activity: 2nd Midterm Exam. It can be improved. A minimum grade is not required. Date of the exam: the date of the realization of the 2nd Midterm Exam, defined by the EPS.

Improvement of Theory Block 2: It consists of performing a 2nd Midterm Exam again. A minimum grade is not required. Date of the exam: the date of the realization of the Improvement Exam, defined by the EPS. The realization of Improvement of Theory Block 2 does not condition the maximum grade achieved in the subject.

Evaluation activities

Acronym	Evaluation Activity	Weight	Minimum Score	Group	Compulsory	Recoverable
EP1	1st Midterm Exam	25%	No	No	No	Yes
EP2	2nd Midterm Exam	50%	No	No	No	Yes
PR1	Practice 1	10%	No	Yes (≤ 2)	No	No
PR2	Practice 2	15%	No	Yes (≤ 2)	No	No
To pass the subject the final score must be ≥ 5.						
Final Score = 0.25 · EP1 + 0.5 · EP2 + 0.1 · PR1 + 0.15 · PR2						

Remarks:

- If the grade obtained in the midterm exam EP2 is > = 5, then this grade may act as an improvement of the first midterm exam EP1.
- The student can choose to improve the midterm exam EP2. The improvement exam is a single written exam. The mark obtained will replace the mark of the two midterm exams of the course.

Alternative evaluation (students who waive continuous evaluation):

Students who have the approval to be evaluated by alternative evaluation (see requirements and procedure in the evaluation regulations) will have to do the following activities.

- **Practice 1** (10%): It cannot be improved. A minimum grade is not required. Delivery date: the date of the realization of the 1st Midterm Exam, defined by the EPS.
- **Practice 2** (15%): It cannot be improved. A minimum grade is not required. Delivery date: the date of the realization of the 2nd Midterm Exam, defined by the EPS.
- **Midterm Exam 2** (75%): It can be improved. A minimum grade is not required. Date of the exam: the date of the realization of the 2nd Midterm Exam, defined by the EPS.
- **Improvement of Midterm Exam 2** (75%): A minimum grade is not required. Date of the exam: the date of the realization of the Improvement Exam, defined by the EPS. The realization of Improvement of Midterm Exam 2 does not condition the maximum grade achieved in the subject.

Bibliography

Algorithms

- Teresa Alsinet, Josep Argelich, Sergi Vila: Programació I. Notes del curs. Eines; Edicions i Publicacions de la Universitat de Lleida.
- Jorge Castro, Felipe Cucker, Xavier Messeguer, Albert Rubio, Lluis Solano, Borja Valles: Curs de Programació. McGraw-Hill, 1992.
- Gilles Brassard, Paul Bratley: Fundamentos de Algoritmia. Prentice Hall, 1997.
- Luis Joyanes: Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos. McGraw-Hill, 2003.

ANSI C and C++

- Harvey M. Deitel, Paul J. Deitel: Cómo Programar en C/C++. Prentice-Hall, segunda edición, 2002.
- Bjarne Stroustrup: Programming: Principles and Practice Using C++. Addison Wesley, 2008.
- Luis Joyanes: Programación en C++. McGraw-Hill, 2006.